Blocking Social Contagions via Dominating Sets Using a Modified Integer Linear Program Formulation

Robert Chen Bao[1], Chris J. Kuhlman[2], and S. S. Ravi[3]

[1] Dept. of Computer Science, University of Virginia, Charlottesville, VA, USA
cb5th@virginia.edu
[2] Advance Research Computing, Virginia Tech, Blacksburg, VA, USA
ckuhlman@vt.edu
[3] Biocomplexity Institute, University of Virginia, Charlottesville, VA, USA
ssravi0@gmail.com

Abstract. Researchers have modeled contagion processes on social networks for wide ranging applications, including spreading of epidemics, financial defaults, actions such as joining social media sites, and rumors. So, too, researchers have developed a host of intervention methods to control harmful contagions on networks; among these approaches are node and edge removal, separating network communities, altering contagion properties, and introducing a second competing contagion. In this work, minimum dominating sets are used to identify blocking nodes—nodes that do not contract a contagion and therefore also do not assist in transmitting it. A novel, generalized method that utilizes integer linear programming to determine exact minimum dominating sets (which is an NP-hard problem) has been developed for a subgraph of *any* social network for *any* combination of covering distance and coverage requirement. Three social networks are used to understand and evaluate (i) the method itself and (ii) the efficacy of the blocking nodes that the method produces to stop contagion spread.

Keywords: Social networks, Contagion, Blocking, Dominating sets, Integer linear programs

1 INTRODUCTION

1.1 Background and Motivation

Social contagion processes in populations have been studied for some 65 years, going back to at least 1955 [22]. Early modeling works of these processes include [16], and continues to this day. Contagions of interest include diseases, rumors, emotions, news, hashtags, joining a group or club, insolence and financial assets, e.g., [14, 20, 29]. A common modeling approach is to represent the people of a population as a set V of vertices, and pairwise interactions between vertices as a set E of edges, thus generating a social network $G(V,E)$. The contagion then propagates from *activated* (i.e., contagion possessing) vertices to *non-activated* vertices through the edges of the network. In this way, non-activated vertices may become activated. Various contagion models are employed across works, such as the susceptible-infectious-recovered (SIR) model and

its variants [21] and threshold models [16]. The Granovetter threshold model is used in this work and is described in Section 1.2.

Since many contagions are undesirable, much work has been done on blocking the progression of contagions. There are various methods for doing this, including using blocking nodes or blocking edges [2, 30, 33] to effectively alter the network structure, making it more difficult for the contagion to spread; using a second ("good") contagion to overtake a first ("undesirable") contagion [6]; and selecting groups of nodes as blocking nodes, based on their attributes, such as age [35]. Illustrations of current works include disease control [32], inhibiting rumors [17], blocking news spreading [6], and halting misinformation [13].

In this work, we introduce a novel approach using dominating sets for identifying blocking vertices in a graph and using these to thwart contagion spreading in social networks, i.e., these nodes will be removed from G. Only one other work has used dominating sets in this way [3] and we differentiate our work from that one in Section 2.

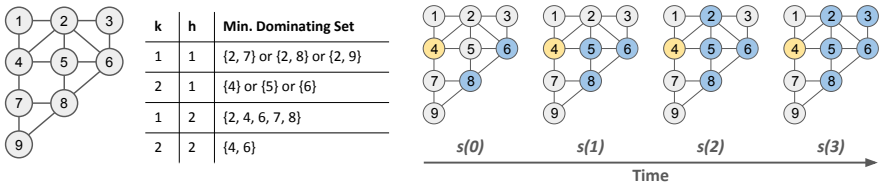
1.2 Overview of Dominating Sets and Contagions

A **dominating set** (DS) of a graph $G(V,E)$ is a set $S_D \subseteq V$ of vertices (nodes) of G such that within a geodesic distance of k (called the **domination distance**) of each $v_i \in V$, there are at least h nodes, including possibly v_i itself (h is called the **coverage requirement**), that are in the dominating set.[4] The typical goal with dominating sets is to find an S_D of G, denoted S_D^{min}, that has a minimum number of nodes, referred to as a **minimum dominating set** (MDS). Computing an MDS for a graph G is an NP-hard problem [12], i.e., it is computationally intractable.

Figure 1a contains a graph and several MDSs, for different values of k and h. For example, for $k = 2$ and $h = 1$, S_D^{min} can be $\{v_4\}$, $\{v_5\}$, or $\{v_6\}$ because all nodes must be covered at least once, and all nodes of G are within distance-2 (the domination distance) of each of v_4, v_5, and v_6. Note how $|S_D^{min}|$ increases for $k = 1$ and $h = 2$, compared to that for $k = 1$ and $h = 1$, because each node must be dominated by two nodes in the former case, as opposed to one in the latter case. The cases $[k,h] = [1,1]$ and $[2,1]$ each demonstrate that an MDS need not be unique.

Figure 1b shows contagion dynamics of the type studied in this work. The network is the same 9-node graph as in Figure 1a. The contagion model is a threshold $\theta = 2$ contagion [16]; that is, at time t, a non-activated node will be activated if at least θ of its neighbors are already activated; else it will remain non-activated. For the initial conditions at time $t = 0$, nodes v_6 and v_8 are selected as the seed nodes, (i.e., they are activated at the beginning of the simulation, driving the contagion to spread). Using a $[k,h] = [2,1]$ dominating set as the blocking set, the DS algorithm selects v_4 as the blocking node. In this case, the contagion will eventually spread to three additional nodes, namely $\{v_2, v_3, v_5\}$. However, it will not activate $\{v_1, v_4, v_7, v_9\}$, as v_4 blocks a sufficient number of contagion pathways to them. Note that if node v_4 is not designated as a blocking node, the contagion would propagate to all nine nodes. So the single blocking node causes only $5/9 = 0.56$ fraction of nodes to activate. Our goal is to study the effectiveness of blocking nodes, selected from MDSs, to halt contagions.

[4] The standard definition of a dominating set is for the case where $k = h = 1$.



(a) (b)

Fig. 1: (a) A 9-node connected graph and MDSs for different domination distances k and coverage requirements h. (b) Contagion spreading time history for a $\theta = 2$ contagion, given $S_D^{min} = \{v_4\}$. The yellow node v_4 is the blocking node (for a $[k,h] = [2,1]$ MDS) and nodes v_6 and v_8 in blue are seed nodes for configuration $s(0)$.

1.3 Novelty and Contributions

A technical novelty of this work is that it is the first to use true MDSs, generated on realistic social networks, to block contagions (an earlier work [3] uses a heuristic for computing approximate MDSs; here we use an exact method). A second novelty stems from the fact that we are interested in studying $[k,h]$ MDSs, for any $k \geq 1$, $h \geq 1$. Exact $[k,h]$ MDSs do not exist for all graphs G. An example is in Figure 2, where a large, arbitrary subgraph J is connected to seven other nodes. If $[k,h] = [2,8]$ is specified, then the coverage requirement for node v_6 cannot be satisfied because there are only $7 < h$ nodes in the closed distance-k neighborhood of v_6, namely nodes v_2 through v_8. A similar result holds for v_3, which can be covered only $6 < h$ times within distance $k = 2$. Thus a $[2,8]$ MDS does not exist for this graph. With no MDS, blocking nodes cannot be determined and evaluated, so we must be able to determine $[k,h]$ MDSs.

With this motivation, a procedure is devised whereby an approximation to G^k is obtained by using the $(h-1)$-core of G^k. An exact $[k,h]$ MDS can be computed on the $(h-1)$-core of G^k, by definition. (The ***m*-core** of a graph G, denoted $H(G,m)$, is the subgraph of G such that every node v_i in $H(G,m)$ has degree $d_i \geq m$ [31]).

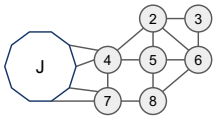


Fig. 2: Graph with subgraph J connected to seven additional nodes through edges to nodes v_4 and v_7. A $[k,h] = [2,8]$ MDS does not exist for this graph. See text for the explanation.

The contributions of this work are:

1. Methodology for computing blocking nodes based on MDSs. A methodology, denoted ILP-MDS, has been developed for computing an exact MDS of a graph G for any dominating distance $k \geq 1$ and coverage requirement $h \geq 1$. Contagion blocking nodes are then selected from the MDS. The process includes, for a specified G, k, h, and blocking budget b the following steps: computing G^k; determining the $(h-1)$-core

of G^k, $H(G^k, h-1)$; computing an *exact* MDS on $H(G^k, h-1)$ using an integer linear program (ILP) method; ranking the nodes of the MDS; and selecting the b highest ranked nodes as the blocking set for contagion simulations. See Section 3.

2. An understanding of the ILP-MDS method. The ILP-MDS methodology is evaluated on three social networks from [24]. For example, since we use $H(G^k, h-1)$ as an approximation of G^k, so that we can compute an MDS on H, we would like for $H(G^k, h-1)$ to have as many nodes of G^k as possible. We demonstrate how the numbers of nodes of the $(h-1)$-core of G^k increase as k goes from one to two, and how they decrease with increasing h. We also compute the number of connected components of $H(G^k, h-1)$ and find that they always have a giant component. Finally, a somewhat surprising result for all three graphs and over a wide range in blocking budget b, is that the degree-sum of all blocking nodes in G is relatively constant for coverage requirement in the range $3 \leq h \leq 7$. This means that the number of transmission pathways (i.e., graph edges) effectively removed from the networks by the blocking nodes is relatively constant, and independent of h. This result is reflected in the efficacy of blocking nodes: as h increases from 3 to 7, the blocking performance for complex contagions, i.e., for threshold $\theta \geq 2$ [7], is relatively independent of h. But this result does not hold for simple contagions, where all node v_i thresholds $\theta_i = 1$, thus illustrating another difference between simple and complex contagion [7]. See Section 4.

3. Efficacy of MDS-based blocking nodes to halt contagion. The efficacies of the resulting blocking node sets are studied in terms of how much they reduce the spread of Granovetter-type threshold contagions [16] in social networks, compared to the spreading with no blocking nodes. For example, it typically takes numbers b of blocking nodes to be on the order of 1000s of nodes (roughly 2000-4000 nodes) or more in order to massively reduce contagion spread (e.g., to reduce contagion spreading to near zero), because our testing/evaluation is onerous: for each set of test variable values, we run 100 simulations with 100 different seed sets. This requires a single blocking node set to block contagion from a wide range of initiation sites. We demonstrate that the classic definition of an MDS, which is the case $k = 1$ and $h = 1$, does not provide the best blocking performance, by a wide margin. Hence, it is critical for effective contagion blocking to devise procedures to compute generalized MDSs, for any $k \geq 1$ and $h \geq 1$, as is done in this work. See Section 5.

4. Comparison of ILP-MDS against the HDH in blocking contagion. The quality of the blocking method is compared against a common standard approach called the high degree heuristic (HDH) studied in many works, e.g., [2, 8, 33]. For a budget b, the b nodes with the greatest degrees are the blocking nodes. It is simple to implement and is highly effective. The result of the comparison is nuanced: in some cases the ILP-MDS method is better than HDH; in some other cases, HDH is more effective; and in still other cases, the two methods generate similar results. Examples of each are provided. See Section 5. Also, while not detailed owing to space reasons, we have verified that the blocking sets from ILP-MDS and HDH are significantly different. Jaccard Index I_J values for the two types of blocking sets are in the range 0.4 to 0.6.

5. Blocking efficacy of the method does not suffer from removing many nodes from G in producing $H(G^k, h-1)$. Overall, a surprising result comes from generating the subgraph $H(G^k, h-1)$ of G^k. That is, one might expect that as the number of nodes

in $H(G^k, h-1)$ becomes a small fraction (e.g., 0.2 to 0.3) of those in G (and G^k), that therefore the MDSs of $H(G^k, h-1)$ and G^k would be very different, and that the contagion blocking performance of the nodes chosen from the MDS of $H(G^k, h-1)$ would suffer because only a small fraction of nodes of G is being considered. We find that this is not the case; that in fact, the blocking nodes generated from the MDS of $H(G^k, h-1)$ when $H(G^k, h-1)$ has only a small fraction of nodes from G, are superior. This is because the nodes that remain from G in $H(G^k, h-1)$ are not only of high degree, but also connected to other high degree nodes.

Paper organization. The remainder of the paper is organized as follows. Section 2 contains related work. Section 3 presents the models and analysis procedures. Section 4 contains network structural results from the procedures. Section 5 provides contagion blocking results. Conclusions and future work are in Section 6.

2 RELATED WORK

Studies of contagion processes, blocking methods, and applications were presented in Section 1.1 and are not repeated.

One work on contagion blocking requires special attention. In [3], $[k,h]$ MDSs are used to select blocking nodes. That approach uses a greedy heuristic that iteratively adds to a node set the node that covers the greatest number of nodes whose coverage requirements are not yet met, breaking ties arbitrarily. An approximate MDS may be computed, but in fact even an approximate MDS is not required. The heuristic operates on the given graph G, with no modification based on $[k,h]$ values. Here, in contrast, G is transformed into an $(h-1)$-core, $H(G^k, h-1)$, of G^k, and the MDS algorithm is run on $H(G^k, h-1)$. Furthermore, a true $[k,h]$ MDS must be computed on $H(G, h-1)$ in this work. Finally, we analyze here the MDS-generating process, which was not done in [3].

Attention is now turned to works on dominating sets. For any $k \geq 1$ (and fixed $h = 1$), finding a distance-k MDS is an NP-Complete problem [12, 15, 19]. Thus, there is much work on heuristics to generate approximate or near MDSs, e.g., [11, 25, 27, 28]. Algorithms for partial dominating sets [4], where a specified fraction of nodes in G must be dominated, are provided in [26, 27].

For particular graph classes, there exist works for scale-free networks [27, 28], preferential attachment networks [9], Erdos-Renyi (ER) random graphs [10], dense ER graphs [34], and random regular graphs [5, 10]. Sizes of partial dominating sets for scale-free networks are studied in [27].

3 Models of Our Dominating Set and Blocking Methodologies

3.1 Overview

The models used in this work and the workflow are given in Figure 3. The steps and models below in Section 3.2 are numbered relative to the boxes in this figure.

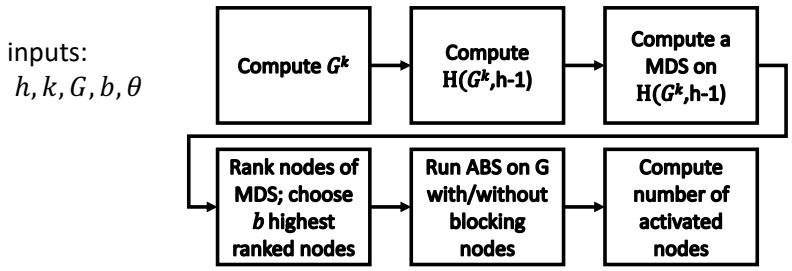


Fig. 3: Process for computing a MDS of a graph G based on DS parameters k and h. The number b of blocking nodes is the greatest ranked b nodes from the MDS; these are used as initial conditions in simulations. The number of activated nodes, i.e., the number of nodes that contract the contagion, is used to quantify the efficacy of the b nodes from the MDS.

3.2 Models

Step 1: Compute graph G^k. Given a domination distance k for the dominating set algorithm, and an original graph G, we determine G^k. The graph G^k is computed from G in the following fashion: for each pair of nodes $\{u,v\} \in V$ of G such that the geodesic distance δ between these nodes is at most k, i.e., $\delta(u,v) \leq k$, place an edge between nodes u and v in G^k. *Motivation:* For general $[k,h]$ MDS, forming G^k means that nodes at distance-k in G are at distance-1 in G^k; thus, software libraries can be used to compute MDSs on G^k.

Step 2: Compute the $(h-1)$-core of G^k. Given a graph G^k and a covering requirement h for the dominating set algorithm, we compute the $(h-1)$-core of G^k. Accordingly, the output from this step is the subgraph $H(G^k, h-1)$. *Motivation:* A $[k,h]$ MDS on G may not exist. Refer to Section 1.3 and Figure 2 for an example. However, by definition of an m-core, a $[k,h]$ MDS exists for $H(G^k, h-1)$, for any $k \geq 1$ and $h \geq 1$. Hence, we use an exact MDS on $H(G^k, h-1)$ as a proxy for an MDS of G.

Step 3: Compute a MDS on $H(G^k, h-1)$. The ILP formulation for computing an MDS of a graph G is as follows. Note that V_H and E_H are the sets of nodes and edges, respectively, of $H(G^k, h-1)$, with $n' = |V_H|$.

Variables: For each node $v_i \in V_H$, we have a $\{0,1\}$-valued variable $x_i, 1 \leq i \leq n'$. The value of x_i is 1 iff v_i is chosen in the $[k,h]$-dominating set for $H(G^k, h-1)$.
Objective: Minimize $\sum_{i=1}^{n'} x_i$
Constraints:

1. Each node must be covered at least h times in $H(G^k, h-1)$. For each $v_i \in V_H$, we must have
$$x_i + \sum_{\{v_i, v_j\} \in E_H} x_j \geq h, \quad 1 \leq i \leq n'.$$
2. Each x_i is a $\{0,1\}$-valued variable: $x_i \in \{0,1\}, 1 \leq i \leq n'$.

Gurobi [18] is used to perform the computations for the above ILP formulation and output an optimal MDS.

Step 4: Rank nodes of an MDS on $H(G^k, h-1)$. Given an MDS, the nodes of the set must be ranked from greatest priority to least priority. *Motivation:* The nodes of an MDS are not naturally ordered. For a given blocking budget b (i.e., the number of blocking nodes for a contagion process), the top-ranked b nodes of the MDS are used as blocking nodes, so a criterion is needed to rank the nodes. In this work, the nodes of an MDS are ranked by their degree in the original graph G, greatest to least, with ties broken arbitrarily.

Step 5: Run agent-based contagion simulations with and without blocking nodes. Simulations of contagion dynamics are run using a discrete dynamical systems (DDS) formulation. A formal description of DDS is provided below. Illustrative dynamics were provided in Section 1.2.

Discrete dynamical system for running agent-based simulations. We model the propagation of social contagions over a social network using discrete dynamical systems [1]. Let $\mathbb{B}$ denote the Boolean domain $\{0,1\}$. Each node $v_i \in V$ is assigned a state 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; an unactivated node has not received the information and consequently cannot pass it on. It is assumed herein that once a node reaches state 1, it cannot return to state 0; this is referred to as a "progressive system" [23]. A **Discrete Dynamical System** (DDS) $\mathcal{S}$ over $\mathbb{B}$ is specified as a triple $\mathcal{S} = (G, \mathcal{F}, R)$, where

1. $G(V,E)$ is an undirected graph with $n = |V|$ nodes, representing the social network on which the contagion propagates;
2. $\mathcal{F} = (f_1, f_2, \ldots, f_n)$ is an ordered collection of functions, with f_i denoting the local vertex 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 distance-1 neighbors in G; and
3. R is the update scheme that specifies the order in which the f_i are executed. Examples are synchronous, sequential, and block sequential [1].

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, we model each function f_i, $(1 \leq i \leq n)$ as a θ_i-threshold function [16] for an appropriate nonnegative integer θ_i. Such a threshold function is defined as follows:

(a) 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.
(b) 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.

The next state at time $(t+1)$ of a node v_i, denoted $s_i(t+1)$, is computed as $s_i(t+1) = f_i(s_i(t)[v_i])$ where $s_i(t)[v_i]$ is the sequence of states of node v_i and of all of its distance-1 neighbors in G at t. The sequence has length $d_i + 1$, where d_i is the degree of v_i.

In our DDS, at each time step t, all the nodes compute and update their states *synchronously*. This means that the next state values $s_i(t+1)$ are computed for all nodes based on all node states at time t. A **configuration** $s(t)$ of a DDS at any time t is an n-vector $s(t) = (s_1(t), s_2(t), \ldots, s_n(t))$. A series of configurations, from $t = 0$ to some integer $t_{max} \geq 0$, denoted as $(s(0), s(1), s(2), \ldots, s(t_{max}))$, is called a **forward trajectory**.

Our simulation engine implements these models to perform agent-based simulations (ABSs), which are computations of forward trajectories. Examples were given in Section 1.2.

Step 6: Compute number of activated nodes. For each forward trajectory of a simulation, raw data from simulation output files are used to count the number of activated nodes. These are used in the results Section 5, as was done in Section 1.2.

4 RESULTS: DOMINATING SETS

Results from Steps 1 through 4 of Section 3 are provided in this section. (Results for Steps 5 and 6 are in Section 5.)

4.1 Network Construction and Determination of $H(G^k, h-1)$

Table 1 shows three networks from [24], along with the square G^2 and the cube G^3 of each graph G. Numbers of nodes and edges, average degrees, and diameters are also given. There are up to 0.5 billion edges in G^3.

Table 1: Number of nodes and edges in mined networks [24] and their second and third power graphs, along with average degrees and graph diameters. The diameter is the longest shortest path in a graph.

Name	**#Nodes**	**#Edges** (G)	**#Edges** (G^2)	**#Edges** (G^3)	**Ave. Deg**(G)	**Ave. Deg** (G^2)	**Ave. Deg**(G^3)	**Dia** (G)	**Dia** (G^2)	**Dia** (G^3)
Astroph	17903	196972	4637830	40425870	22.00	518.12	4516.10	14	7	5
Enron	33696	180811	15237618	156994847	10.73	904.42	9318.31	13	7	5
Epinions	75877	405739	36788202	522549279	10.69	969.68	13773.59	15	8	5

Figure 4a provides the fraction of original nodes from G in $H(G^k, h-1)$, i.e., in the $(h-1)$-core of G^k, for the particular case of $k = 1$. These fractions are plotted against $(h-1)$. The curve for the AstroPh network is above those for the other two networks because of its greater average degree, which is roughly twice that of the other two networks. Because an MDS is selected from $H(G^k, h-1)$, and since a (sub)graph with all nodes of G^k will permit choosing nodes of an MDS from all graph nodes, intuitively, therefore, one might want the number of nodes of $H(G^k, h-1)$ to be a large fraction of those in G^k.

Figure 4b shows data for the same three networks, but now $k = 2$. The curves are now much higher—the fractions of nodes of G in $H(G^k, h-1)$ are much greater—because G^2 is being used, and Table 1 shows how the average degrees increase by $20\times$ to $90\times$ in going from G^1 to G^2.

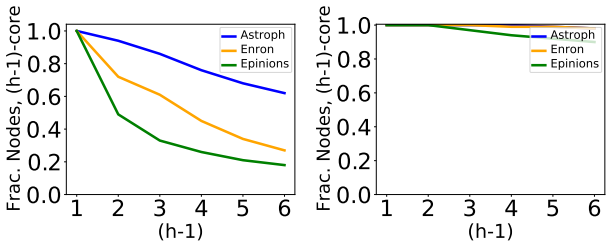


Fig. 4: Fraction of nodes in $H(G^k, h-1)$ of each network G^k (and G) as a function of $(h-1)$, for various h with (a) $k=1$ and (b) $k=2$.

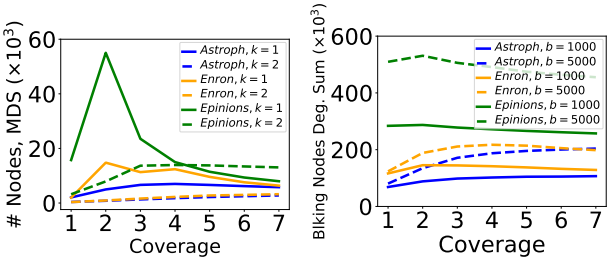


Fig. 5: (a) Number of nodes in MDSs for the three networks (in different colors) in G^k for $k=1$ (solid) and for $k=2$ (dashed). The numbers of nodes in MDSs increase and then essentially plateau as a function of coverage requirement h. (b) Sum of the degrees in the blocking nodes for the three networks G for $k=1$ and varying h (on the x-axis) for blocking budgets $b=1000$ and 5000 nodes. Over the range $3 \leq h \leq 7$, the sum of the degrees of all blocking nodes in G is relatively constant in h.

4.2 MDSs on $H(G^k, h-1)$ and Blocking Nodes

Figure 5a shows the number of nodes in the MDSs that are computed with Gurobi in Step 3, for each of the three networks, for a range of combinations of $[k,h]$. The curves are not monotonic in h. For each network and for a given k, the size of the MDS increases as h increases from 1 to about 2 or 3. This is explained as follows. For a given node v_i and a fixed k value, as h increases, the density of dominating nodes in the k-neighborhood of v_i increases; this follows immediately from the definition of a $[k,h]$ MDS because a node has to be covered more times. As h increases beyond about 3, the number of nodes in an MDS slightly decreases or remains about constant. This suggests, for $h \geq 3$, that each dominating node v_ℓ is dominating more $v_j \in V$, in the k-neighborhood of v_ℓ. This ability of some dominating nodes to dominate many other nodes results in decreasing or relatively constant sizes of MDSs as h increases.

The diameter values in Table 1 also reinforce this last point. Since the density of edges increases in G^k as k increases (and the diameter of G^k decreases by a factor of about two as k increases from 1 to 2), a dominating node has more neighboring nodes, and hence can assist in meeting more nodes' coverage requirements. This is why the dashed curves for $k = 2$ are mostly markedly below the solid ones for $k = 1$ in Figure 5a; except for Epinions and $h > 4$.

Nodes from an MDS are ranked according to Step 4 of Section 3.

Figure 5b offers a somewhat surprising result on the effect of blocking nodes, with respect to structure of the graph (specifically, contagion pathways) and coverage requirement. Note that an edge of a graph is effectively rendered useless in propagating contagion if at least one of its nodes is a blocking node, for in this situation, no contagion can pass to or from an inactive node. Figure 5b shows the sum of the degrees of all blocking nodes in G for $k = 1$, for all three networks, for blocking budgets $b = 1000$ and 5000 blocking nodes, as a function of coverage requirement h. These degree sums are computed on the *original* graphs G, since these are the graphs on which contagion will spread. The degree sums in the figure are relatively constant (some curves slightly increase, others slightly decrease) over an appreciable range in coverage: $3 \leq h \leq 7$. This implies, to a first approximation, that the number of edges in the graphs G that are blocked (i.e., have at least one dominating node forming that edge) is roughly the same, irrespective of h, over the range $3 \leq h \leq 7$.

It is surprising in that one might reasonably expect the curves in Figure 5b to increase as h increases. However, the description of Figure 5a explains these results: as h increases from 3, the number of nodes in an MDS is relatively independent of h. Nonetheless, the results are interesting because they explain some results in the next section: *blocking efficacy asymptotically plateaus as h increases, for complex contagions*; see $\theta = 3$ results in Figures 7a and 7b.

5 RESULTS: EFFICACY IN BLOCKING CONTAGIONS USING MDS

The simulation model is given in Section 3, Step 5. Illustrative simulations are given in Section 1.2. Simulation parameters are given in Section 5.1 below, and Section 5.2 contains agent-based simulation (ABS) contagion blocking results.

5.1 Simulation Scenarios and Process

Each simulation is a collection of 100 forward trajectories. Each forward trajectory of a collection has the same initial conditions, except that the composition of the seed node set is different for all 100 trajectories (but the sets are all of the same size n_s). Different seed node sets more thoroughly test the ability of the blocking nodes to halt contagion that may start from different locations in the network. Simulation parameters are: (i) network G, (ii) number $n_s > 0$ of seed nodes, (iii) the dominating set parameters $[k,h]$, with $k \geq 1$ and $h \geq 1$, (iv) the number $b \geq 0$ of blocking nodes (i.e., blocking budget), and (v) the threshold $\theta_i \geq 1$ assigned to each node $v_i \in V$; in this work, all nodes are assigned the same threshold in a simulation.

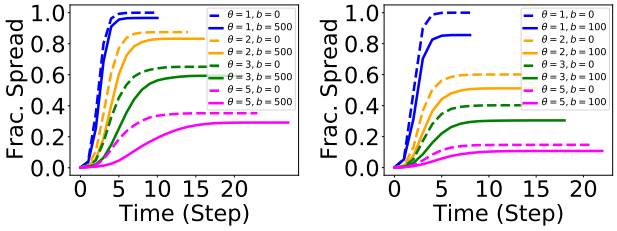
A fixed set of all parameters constitute one simulation. Each forward trajectory of a simulation starts at time $t = 0$, where the configuration $s(0)$ is known, and runs through a maximum time $t_{max} = 50$ time steps. At each time step $t \geq 1$, the contagion dynamics are computed, to determine which nodes of the graph contract the contagion at that time. Results below are averages over 100 forward trajectories of a simulation.

5.2 Simulation Results

Here, we provide results from contagion simulations and analyze the efficacy of the MDS blocking method. A blocking method's effectiveness increases as the number of activated graph nodes decreases. Hence, for the plots in this section, the lower the curve (i.e., the lesser the y-axis value for a particular x-axis value), the more effective is the blocking method.

Time histories for contagions with and without blocking Figure 6 contains time histories of *cumulative* fractions of activated nodes for AstroPh and Enron. In each plot there are pairs of curves: one dashed curve (where no blocking nodes are used) and one solid curve (where blocking nodes are used). The solid curves are below the dashed curves for the different colors, indicating that the blocking nodes reduce contagion spread. The different colors correspond to different node threshold θ values. As threshold increases—making it more difficult for the contagion to propagate—the curves show lesser fractions of activated nodes.

The final value on the y-axis for each curve is the greatest fraction of nodes reaching state 1, i.e., is the greatest fraction of activated nodes, and is referred to as *final fraction* or *final spread fraction*. These values will be used in subsequent plots.



(a) (b)

Fig. 6: Time-histories of cumulative fraction of activated nodes (i.e., spread fraction) in ABSs for (a) AstroPh with $n_s = 10$, and $b = 500$, and (b) Enron with $n_s = 10$, and $b = 100$. The blocking results ($b > 0$, solid curves) are for the ILP-MDS algorithm with $[k,h] = [1,7]$, for comparisons with no-blocking baselines ($b = 0$, dashed curves). Thresholds $\theta = 1$ to 5 are used in simulations. Blocking "lowers" each curve, indicating that blocking inhibits contagion.

Table 2: Effect of domination or covering distance k on contagion spread size (i.e., final spread fraction) for $k = 1$ and $k = 2$ with varying numbers n_s of seed nodes. Each row corresponds to a given network with a fixed blocking budget b, coverage requirement h, and threshold θ. Each cell value represents the final (spread) fraction for the given condition. Results over all analyses show that $k = 2$ sometimes match the performance from $k = 1$, but generally underperform.

Network and Condition	$k=1$			$k=2$		
	$n_s = 10$	$n_s = 100$	$n_s = 1000$	$n_s = 10$	$n_s = 100$	$n_s = 1000$
AstroPh ($b = 5000$, $h = 7$, $\theta = 1$)	0.49	0.50	0.56	0.74	0.74	0.76
Enron ($b = 5000$, $h = 1$, $\theta = 1$)	0.45	0.52	0.74	0.71	0.75	0.90
Epinions ($b = 5000$, $h = 7$, $\theta = 1$)	0.49	0.53	0.65	0.50	0.53	0.65

Effects of covering distance k in the ILP-MDS Algorithm Table 2 lists final spread fractions as a function of the covering distance or domination distance k for all three networks with different simulation conditions. For Epinions, the effect of k is small. For Enron and AstroPh, however, the spread fraction increases markedly with increasing k. Results are not available for $k = 3$ because as Table 1 indicates, the number of edges in G^k for $k = 3$ is quite large (hundreds of millions of edges for two of the three graphs) and computing dominating sets on these graphs, even with high performance computing (HPC) cluster hardware nodes, is challenging. For example, it took over ten days to compute the diameter for G^3, i.e., Dia.(G^3), of Epinions on an HPC compute node.

Effects of coverage requirement h in the ILP-MDS Algorithm The effect of coverage requirement h on the final spread fraction is given in Figure 7 for all three networks. Figures 7a and 7b illustrate that the final spread fraction decreases monotonically as h increases, for two networks, with complex contagion, i.e., for $\theta \geq 2$ [7]. This makes sense in light of Figure 5b, since as h increases from 1 to about 3, the sums of the degrees of the blocking nodes increase (see dashed curves for $b = 5000$). Thereafter (i.e., for $h \geq 3$), the curves in Figure 5b do not increase much (in the case of Enron, it falls slightly) and the curves plateau. The curves for AstroPh in Figure 7 show more spreading simply because it has a larger average degree than does Enron—the ratio of threshold $\theta = 3$ to average degree is much smaller for AstroPh, meaning that there are more pathways for the contagion to spread in AstroPh.

Figure 7c is surprising at first in that the final spread fraction is not monotonic in the coverage requirement. The conditions in these simulations are identical to those in Figure 7b except the threshold has now been reduced to $\theta = 1$, and is therefore a simple contagion [7]. This means that fewer pathways (e.g., activated neighbors) are needed to activate a $\theta = 1$ contagion. All three networks exhibit this behavior for $\theta = 1$. Thus, the pathway argument described as part of Figure 5b does not hold for simple contagions. This illustrates the dependence of results on combinations of input parameters (here, threshold, average degree, and coverage requirement).

Note that three of the four plots demonstrate that the standard definition of an MDS, with $k = 1$ and $h = 1$, does not result in the best blocking performance, and hence we need to evaluate $k \geq 1$ and $h \geq 1$.

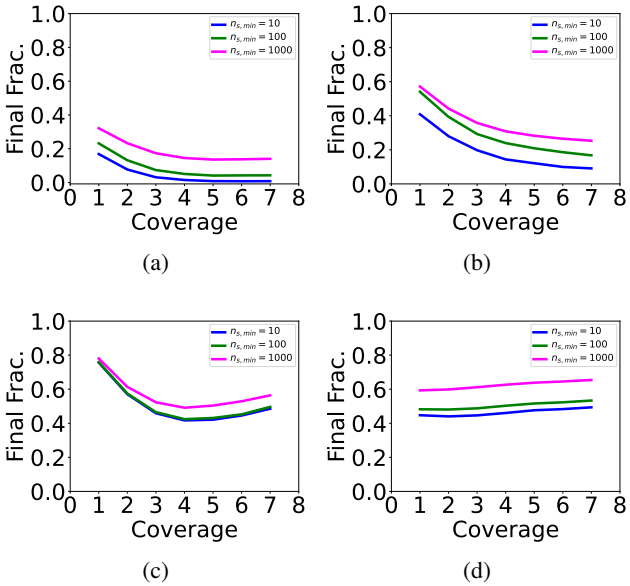


Fig. 7: Effect of coverage requirement h on final spread fraction. (a) Enron with $b = 5000$, $k = 1$, and $\theta = 3$. (b) AstroPh with $b = 5000$, $k = 1$, and $\theta = 3$. (c) AstroPh with $b = 5000$, $k = 1$, and $\theta = 1$. (d) Epinions with $b = 5000$, $k = 1$, and $\theta = 1$. The first two plots show monotonically decreasing final spread fractions as h increases. The third and fourth plots exhibit non-monotonic and non-decreasing behaviors, respectfully.

Effect of number b of blocking nodes (blocking budget) The effect of blocking budget b on the final spread fraction is given in Figure 8 for all three networks. (Please note that we use Figure 8 for this subsection and the next, for space reasons; disregard the HDH blue curves in the plots at this time.) In the first two plots, for small blocking budgets, on the order of $b = 1000$ to 2000, or less, the cyan curves for $[k,h] = [1,1]$ demonstrate the most effective blocking. Beyond this point, the cyan curves plateau, indicating no more effect of b. However, the flat portions of the cyan curves denote that there are no more blocking nodes to add (all nodes of the MDS are being used as blocking nodes). This is confirmed in Figure 5a, which shows 2000 nodes in the MDS for $[k,h] = [1,1]$ for AstroPh. Figure 8a shows a consistent trend: as h increases from one, the curves are lower, but only up to a point, before they plateau and their performance is superseded at greater b by curves with larger h, for $h \leq 5$. For h increasing from 5 to 7, the same behavior as in Section 5.2 for simple contagions is observed: the spread fraction increases with increasing h.

In contrast to the above results, in Figure 8b, the curves are lower as h increases, with little effect of different regimes of b. This is because the simulation here is for a complex contagion.

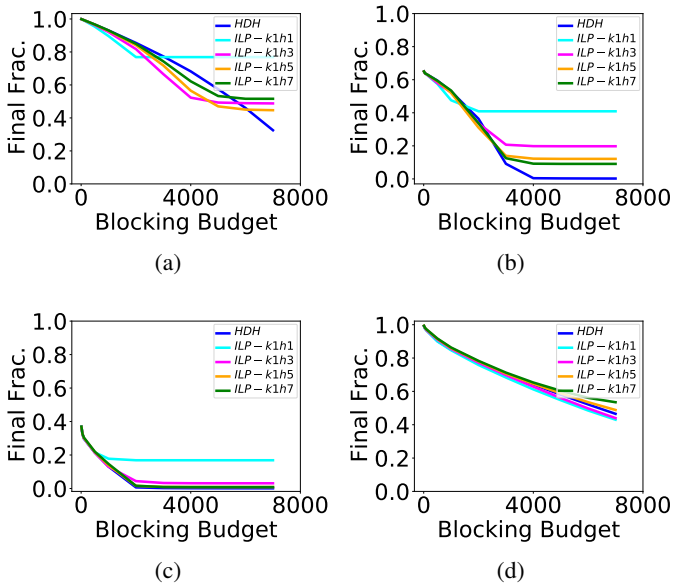


Fig. 8: Comparison between performance of DS Algorithm (ILP-MDS) and High-Degree Heuristic (HDH). (a) AstroPh with $n_s = 500$ and $\theta = 1$. (b) AstroPh with $n_s = 10$ and $\theta = 3$. (c) Enron with $n_s = 10$ and $\theta = 3$. (d) Epinions with $n_s = 500$ and $\theta = 1$. The plots demonstrate that ILP-MDS performs better than HDH does, see (a) and (d); that HDH performs better than ILP-MDS does, see (b); and that the two methods are comparable, see (c).

In Figure 8c, the curves for all combinations of $[k,h]$, except for $[1,1]$ are essentially the same, due to the fact that Enron has roughly one-half the average degree of AstroPh. Thus, it is harder to propagate $\theta = 3$ contagion in Enron, as stated above.

Figure 8d, for Epinions, shows that the curves are all roughly comparable in the regime $0 \leq b \leq 4000$. Thereafter, the curves differentiate. Because this is a threshold $\theta = 1$ contagion, the curves *increase* as h increases from three, to five, to seven, thus producing a variant of the surprising behavior in Figure 7c.

Comparison of Our ILP-Based Dominating Set Method to the HDH In this section, we compare ILP-MDS solutions to high degree heuristic (HDH) solutions. For a given blocking budget b, the HDH simply uses as blocking nodes the b nodes with greatest degrees in G, breaking ties arbitrarily. As stated in Section 1.3, the HDH is a common standard of comparison for contagion blocking methods since it is simple to implement and often quite effective. We again use Figure 8, but now we address the blue HDH curves.

The Figure 8 plots were chosen to illustrate three types of behaviors: (i) the ILP-MDS is more effective in blocking contagions (at least over the great majority of b values): Figures 8a and 8d; (ii) the HDH is more effective in blocking contagions (at

least for larger b): Figure 8b; and (iii) the two methods are comparable in blocking contagions: Figure 8c. Jaccard Index I_J values for the HDH- and ILP-MDS-generated blocking node sets indicate that the two methods generate different sets; I_J values range from 0.4 to 0.6.

6 CONCLUSION AND FUTURE WORK

This work presents a procedure to compute exact minimum dominating sets (MDSs) of a subgraph of a graph G^k for any algorithm parameter set $[k,h]$, with $k \geq 1$, $h \geq 1$ and graph G. Blocking nodes are selected from MDSs and are used in agent-based simulations to study the efficacy of this contagion blocking approach. See Section 1.3 for our contributions. Future work includes computing different MDSs to determine the sensitivity of results to these MDSs, if they exist. An interesting new avenue for future work is to build a neural network to determine which combinations of $[k,h]$ give the best blocking results for different conditions. Often $[k,h] = [1,7]$ give best results, but this is not always the case.

ACKNOWLEDGMENT

This work has been partially supported by UVA Strategic Investment Fund number SIF160 and NSF Grants CMMI-1916670 (CRISP 2.0) and OAC-1916805 (CINES).

References

1. Adiga, A., 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)
2. Albert, R., Jeong, H., Barabasi, A.: Error and attack tolerance of complex networks. Nature **406**, 378–382 (2000)
3. Bao, R.C., Hancock, M., et al.: Using dominating sets to block contagions in social networks. In: ASONAM. pp. 22–25 (2022)
4. Berge, C.: Theory of Graphs and its Applications. Methuen (1962)
5. Biro, C., Czabarka, E., Dankelmann, P., et al.: Remarks on the domination number of graphs. Bull. Inst. Combin. Appl **64**, 7 pages (2012)
6. Budak, C., Agrawal, D., El Abbadi, A.: Limiting the spread of misinformation in social networks. In: WWW. p. 665–674 (2011)
7. Centola, D., Macy, M.: Complex Contagions and the Weakness of Long Ties. Am. J. Sociology **113**(3), 702–734 (2007)
8. Centola, D.: Failure in complex social networks. Journal of Mathematical Sociology **33**, 64–68 (2009)
9. Cooper, C., et al.: Lower bounds and algorithms for dominating sets in web graphs. Internet Mathematics **2**, 275–300 (2005)
10. Erdos, P., Renyi, A.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci. **5**, 17–61 (1960)
11. Eubank, S., Kumar, V.S.A., et al.: Structural and algorithmic aspects of massive social networks. In: ACM-SIAM Symposium on Discrete Algorithms. pp. 718–727 (2004)

12. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
13. Ghoshal, A.K., Das, N., et al.: Influence of community structure on misinformation containment in online social networks. Knowledge-Based Systems **213**, 106693:1–12 (2021)
14. Glasserman, P., Young, H.P.: Contagion in financial networks. Journal of Economic Literature **54**, 779–831 (2016)
15. Grandoni, F.: A note on the complexity of minimum dominating set. Journal of Discrete Algorithms **4**, 209–221 (2006)
16. Granovetter, M.: Threshold models of collective behavior. American Journal of Sociology **83**(6), 1420–1443 (1978)
17. Guo, J., et al.: A multi-feature diffusion model: Rumor blocking in social networks. IEEE/ACM Trans. on Networking **29**, 386–397 (2021)
18. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2023), https://www.gurobi.com
19. Henning, M.A.: Distance domination in graphs. In: Haynes, T.W., et al. (eds.) Topics in Domination in Graphs, pp. 205–250 (2020)
20. Herrando, C., Constantinides, E.: Emotional contagion: A brief overview and future directions. Frontiers in Psychology **12** (2021)
21. Hethcote, H.W.: The mathematics of infectious diseases. SIAM Review **42**, 599–653 (2000)
22. Katz, E., Lazarsfeld, P.F.: Personal influence: the part played by people in the flow of mass communications. Free Press (1955)
23. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the Spread of Influence Through a Social Network. In: KDD. pp. 137–146 (2003)
24. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data (Jun 2014)
25. Lovasz, L.: On the ratio of optimal integral and fractional covers. Discrete Mathematics **13**, 383–390 (1975)
26. Molnar, F., Derszy, N., Szymanski, K., Korniss, G.: Building damage-resilient dominating sets in complex networks against random and targeted attacks. Scientific Reports **5**, 1–11 (2015)
27. Molnar, F., Sreenivasan, S., et al.: Minimum dominating sets in scale-free network ensembles. Scientific Reports **3**, 383–390 (2013)
28. Nacher, J.C., Akutsu, T.: Dominating scale-free networks with variable scaling exponent: Heterogeneous networks are not difficult to control. New Journal of Physics **14** (2012), 24 pages
29. Rizzo, A., Pedalino, B., Porfiri, M.: A network model for Ebola spreading. Journal of Theoretical Biology **394**, 212–222 (2016)
30. Saha, S., Adiga, A., et al.: Approximation algorithms for reducing the spectral radius to control epidemic spread. In: SDM. pp. 568–576 (2015)
31. Seidman, S.B.: Network structure and minimum degree. Social Networks **5**, 269–287 (1983)
32. Sheikhahmadi, A., Bahrami, M., Saremi, H.: Minimizing outbreak through targeted blocking for disease control: a community-based approach using super-spreader node identification. Scientific Reports **13**, 14217:1–17 (2023)
33. Tong, H., Prakash, B.A., Eliassi-Rad, T., Faloutsos, M., Faloutsos, C.: Gelling, and melting, large graphs by edge manipulation. In: CIKM. pp. 245–254 (2012)
34. Wieland, B., Godbole, A.P.: On the domination number of a random graph. The Electronic Journal of Combinatorics **8** (2001), 13 pages
35. Zhang, Y., Adiga, A., et al.: Near-optimal algorithms for controlling propagation at group scale on networks. IEEE Transactions on Knowledge and Data Engineering **28**(12), 3339–3352 (2016)