Block Contagions in Social Networks using Dominating Sets

A Technical Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Robert (Chen) Bao

with

Matthew Hancock
Chris J. Kuhlman
S. S. Ravi

May 8, 2023

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Robert (Chen) Bao

Technical advisor: Rosanne Vrugtman, Department of Computer Science

Block Contagions in Social Networks using Dominating Sets

CS4991 Capstone Report, 2022

Robert (Chen) Bao, Matthew Hancock, Chris J. Kuhlman, S. S. Ravi
Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia, USA
cb5th@virginia.edu

ABSTRACT

Contagions—such as viruses, misinformation, and technology—are ubiquitous in real-life; however, many of them are harmful to society. A UVA professor and I propose a novel method of blocking contagions in social networks, using nodes from dominating sets. Our method produces a prioritized list of dominating nodes using a Python program. By “inoculating” these nodes from the contagion, we effectively reduce the contagion’s damage—in many networks, eliminating up to 40% of contagion spread. We also demonstrate the effectiveness of our approach by comparing blocking results with those from the high degree heuristic (HDH), a common standard in blocking studies. Our research will help scholars and policymakers make informed decisions; additionally, the resulting paper will soon be published at a social network conference. In the future, we would like to improve the speed of the algorithm, so it can be applied to large social networks.

1. INTRODUCTION

Real-life contagions like misinformation, technology and viruses, including COVID-19, can be both harmful and damaging. For example, since 2020, COVID-19 and misinformation have led to billions of dollars of losses, and a pronounced decline in people's well-being. To combat these contagions, decision-makers, like governments and social media platforms, need effective ways to contain their spread.

2. RELATED WORKS

Researchers have done extensive work to block contagions spread on social networks. Often, they model each individual in the network as a node. The connections between nodes are called *edges*. Computer scientists have developed numerous approaches based on this node-edge model of networks. Among them, the one most widely used is node removal. With this approach, the blocking algorithm removes or inoculates individual nodes that can contract contagions. Berge (1962) was among the first to suggest eliminating susceptible nodes as a potential solution to reduce contagion spread. Since then, many scientists have adapted this method for specific use cases, like controlling epidemics (Chen, 2018). As an alternative method, other scientists have proposed blocking contagions by removing edges (Tong, 2012). Although less used, this solution is applicable in situations where removing nodes carries a high cost.

We have considered both node and edge removal methods. In the end, we decided to build a novel algorithm based on the node removal method described by Berge, for its ease of implementation and fast performance. Specifically, we developed the algorithm to select nodes to “inoculate” based on

Dominating Sets, thus building defenses against the spreading contagion.

3. PROJECT DESIGN

For this project, I have developed a novel algorithm to block contagions. We also tested its performance using simulations with three well-known network datasets.

3.1 The Dominating Set Algorithm

Our Dominating Set Algorithm (DSA) has the following key steps. First, for each node, determine the set of all of its neighbors. Second, build dominating sets from each node. Identify the node v_j with the greatest number of neighbors c_j. Third, the program writes this pair (v_j, c_j) to the output file of prioritized dominating nodes. The algorithm repeats the steps until the prioritized node count reaches the blocking budget.

3.2 Networks Used in Simulations

To comprehensively test our algorithm, we used three well-known network datasets. They were collected from real-world scenarios; thus, they offer a realistic assessment of DSA's performance. They also come from very diverse sources, ensuring a sufficiently wide analysis scope.

Network	**Type**	**Num. Nodes**	**Num. Edges**	**Ave. Deg.**
AstroPh	collaboration	17903	196972	22.0
Enron	Email	33696	180811	10.7
Epinions	online social	75877	405739	10.6

Table 1: Mined networks and selected properties.

As shown in Table 1, the social networks are AstroPh, Enron, and Epinions. AstroPh stands for AstroPhysics, and it covers collaborations between astrophysicists in authoring papers. Enron, meanwhile, is a communication network. It contains the email interactions between employees of Enron Corporation before its collapse. Lastly, Epinions contains social connection data from Epinions.com, a social site focused on consumer reviews.

3.3 The Simulation Pipeline and Processes

To examine our algorithm in action, we ran simulations with our algorithm on the networks in Table 1. In real life, it is often unrealistic to "inoculate" every individual. Thus, for each simulation, we set a blocking budget, the maximum number of nodes we are allowed to "inoculate." Its value ranges from 10 to 7000. Also, some contagions require multiple contacts to infect an individual. So, we set a different threshold, the number of contacts required for a node to be "infected," for each simulation. The threshold ranges from 1 to 10.

Then, for each network, we ran simulations following the steps in Figure 1. First, we used DSA to generate candidate blocking nodes for each network dataset. Then, we randomly generated 100 sets of seed nodes that will be "infected" at the beginning of the simulation, for each preset seed set size, $n_{s,\ min}$. Next, we selected the blocking nodes based on our budget. We ran simulations for all 100 scenarios. Finally, we computed the fraction of nodes infected in the end and used it to evaluate the algorithm's performance.

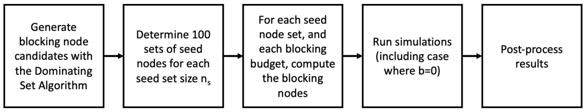

Figure 1: Diagram of Simulation Process. This pipeline is for one network only.

3.4 Challenges

In developing and testing the algorithm, we encountered several major challenges. First, it was challenging to select the most representative network datasets for testing. The conference enforces a strict length limit for papers (within four pages). Thus, it was crucial to use a minimal number of networks to save space, while assessing the algorithm thoroughly. To address the issue, we searched twelve network datasets for the three with the most representative properties As shown in Table 1, the selected networks cover a diverse set of traits, including type, size (measured by the number of nodes), and average degrees. In the end, they were able to show the full spectrum of the DSA's performance through testing.

The second challenge we faced was optimizing the algorithm. The early version of DSA was resource intensive. For example, for large networks like Epinions, the algorithm often consumes over 64 GB of memory. This makes the algorithm unrealistic for real-world applications, where large input networks are common. To optimize the program, I restructured it to incorporate a more efficient hash map data structure. It improved the speed by 2x, which proved to be sufficient.

4. RESULTS

Through simulations, we were able to show that our algorithm does offer a significant defense against contagions, stopping 25% of the spread.

Figure 2 provides time-history results for the Enron graph. Each curve is the fraction of infected nodes as a function of time. There are four pairs of curves, corresponding to four values of threshold. For each pair, one curve is for the case with no blocking nodes, while the other is for the case with *budget = 1000*. The blocking (solid) curve is below the non-blocking (dashed) curve.

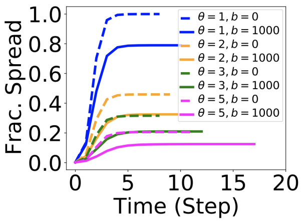

Figure 2: Time-histories of cumulative fraction of infected nodes in the Enron network.
The number of nodes infected at the start is 1000. The blocking results (budget=1000, solid curves) are for the DSA, for comparisons with no-blocking baselines (dashed curves) of the same color.

As shown in Figure 2, the strengthened network finished with 25% fewer infections. Note that Enron is a large network with over 30,000 nodes, and our "inoculated" nodes account for only 3% of its members. Thus, our algorithm had an outsized impact on preventing contagion spreads.

The results show that the DSA can efficiently block contagion spread and it could be an important tool for decision-makers in the future. For example, it may help manage a future outbreak or limit the spread of misinformation on social media. Moreover, as the blocking method will soon be publicly available, this research will help other scientists further their own research, building more effective contagion-blocking methods.

5. CONCLUSION

In this research, we developed a novel algorithm to block contagion spread using networks. We also evaluated its performance using three well-known social network datasets. This project will help policymakers make informed decisions, giving them a strong tool to contain harmful contagions like viruses or misinformation. It will also enable researchers to further their research and create robust tools to counter contagions.

6. FUTURE WORK

Future work includes analyzing the algorithm's blocking performance in detail, and optimizing it for large-scale datasets. We may also explore additional social network datasets, testing the algorithm more extensively.

REFERENCES

[1] C. Berge, Theory of Graphs and its Applications. Methuen, 1962.

[2] 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.

[3] X.-L. Chen, S.-M. Cai, M. Tang, W. Wang, et al., “Controlling epidemic outbreak based on local dynamic infectiousness on complex networks,” *Chaos*, vol. 28, no. 12, pp. 123 105–1–123 105–11, 2018.