One-Shot Attack Heuristics For Viral Containment

Silva A.[1] and Conde B.[1]

[1]Instituto Superior Técnico - Universidade de Lisboa

***Abstract*—Stopping viral spreading in real-world networks is an essential task in life-saving missions such as epidemic containment. Identifying which nodes to attack, or vaccinate, so as to optimize these efforts, is a hard task, especially in low-resource environments. Here we present several one-shot heuristics (i.e. computed once, at the beginning) for selecting which nodes to attack first. We also present and discuss the results of an experimental setup based on the SIS model, where the proposed one-shot heuristics attack real-world networks with the aim of reducing the number of infected nodes. The experimental results show that existing techniques for identifying influential nodes in viral spreading may not translate well into one-shot heuristics. Further, we suggest that one-shot heuristics which depend on structural information of the network are worst than local information oriented heuristics. Our experiments highlight the need for caution when selecting which heuristics to use in a low-resource environment.**

I. Introduction

Identifying influential nodes of real-world graphs in the context of viral spreading is an important research domain. Its importance comes from the ubiquitousness and impact of viral mechanisms, such as the spread of fake information or infectious diseases.

Halting these dynamics is a vital step in life-saving missions such as the one the world has faced during the SARS-CoV-2 pandemic. One obvious way of halting the spread in such contexts is by removing, or vaccinating, influential nodes from the graph; nodes that, when infected, result in a large cascading infection through the rest of the network. By removing these nodes, we do not allowed them to become spreaders, with the aim of reducing or even stopping the spread.

However, techniques for identifying influential nodes often require abundance of information about the network, which is not cheap to obtain in real-world scenarios.

In this work, we consider 7 different one-shot heuristics (i.e. are only computed once, at the beginning of the process) for attacking a network, and simulate the viral process in relation to them. We consider the Susceptible-Infected-Susceptible (SIS) as the compartmental model, and the LFR-Benchmark [1] graph as a base model for representing real-world networks.

We present and discuss the the results of these simulations with multiple percentages of nodes removed by attacks, comparing the presented heuristics between themselves and for different β values of the SIS model.

II. Related Work

Multiple previous work has been done in order to find a network's influential spreaders ([2], [3] argues that the most influential spreaders are nodes belonging and adjacent to the K-Shell with the biggest K; [4] also uses the biggest K-Shell, but adds classification weights to each of the nodes; [5] uses a new Hybrid Centrality measure, called HybridRank, that takes into account the topology of the network; and [6] introduces a new Gravity Centrality measure, based on K-Shell decomposition and Shortest Path size between two nodes). There has also been influential analysis [7] of some of the heuristics discussed in this work, mostly interested on the differences of efficacy of these heuristics in epidemic spreading versus gossip/rumor dynamics. The main contributions of our work is providing a statistical analysis of efficacy of all of these different heuristics on graphs similar to real-life communities, therefore emulating viable, real-world scenarios.

III. Methods

A. SIS

The Susceptible-Infected-Susceptible (SIS) model, represented in Fig. 1, describes a situation of epidemic spreading when the infected individuals, while able to recover from an infection, are never able to gain full immunity to it. This model tends to be very popular in modeling the common cold and influenza, due to its parameter simplicity and computational efficiency.

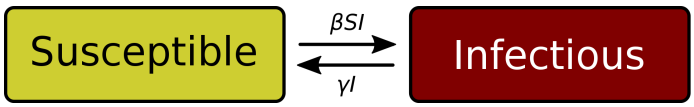


Fig. 1. The state transitions of whole graph in a SIS model. β is the probability of a node in the Susceptible state becoming Infectious, when connected to an Infectious node. γ is the probability of an Infectious node recovering and becoming Susceptible again.

The reason for choosing the SIS model instead of other, more complicated models, such as the Susceptible-Infected-Recovered model (SIR) [8] model, is due to its simplicity: the objective of this work is to test the efficacy of a viral spreading related to the attacks on a network. This means that the accurate final results are not very important: the most important thing is that the removal or non-removal of certain nodes has a significant, predictable effect on the final infection rate. This makes the SIS model the simplest possible model available that is still able to meet our demands.

B. LFR-Benchmark

Benchmark graphs are graphs that are artificially generated, but try to mimic real-world networks. They consist of a

number of sparsely connected communities, where all nodes in a community are densely connected.

Lancichinetti–Fortunato–Radicchi benchmark [1], or LFR-Benchmark, is an algorithm that is able to generate these benchmark graphs with a specific number of communities desired, while accounting for the heterogeneity of node degrees and community sizes. It assumes that both the node degrees and community sizes follow a power law, something that is also normally seen in real life environments. All graphs for this paper were generated with a fixed seed, so confirmation of results is possible and simple. For the LFR-Benchmark generation, we use 3 as the power law exponent for the degree distribution, 1.5 as the power law exponent for the community size distribution, and 13 as the average node degree. The fraction of inter-community edges incident to each node is fixed at 10%. All communities in this graph must have at least 20 nodes. The number of nodes if set to 10^5. We chose these values as they create networks capable of mimicking real-world network behaviors, while still allowing a very fast generation process.

C. One-Shot Attack Heuristics

One-shot attack heuristics are defined by being computed only once, at the beginning of the attack process. This means no information is supplied to them afterwards, even if the network fundamentally changes its composition. We consider the following 7 one-shot attack heuristics:

Random heuristic At each step, a random node is removed from the network. It is used as a baseline, since it is the simplest heuristic, requiring only knowledge about which nodes exist in the network.

Degree centrality heuristic At each step, remove the node with the highest degree centrality still available. The degree centrality is defined as the proportion of nodes connected to the given node.

K-Shell [9] heuristic At each step, randomly remove a node from the K-Shell with the maximum coreness still available. The K-Shell is defined as the set of nodes belonging to the K-Core but not to the (K+1)-Core. A K-Core is is defined as the maximal sub-graph whose nodes' degrees are greater or equal to K.

Betweenness centrality [10] heuristic At each step, remove the node with the highest betweenness centrality still available. The betweenness centrality of a node is defined as the sum of the proportion of shortest paths between any other two given nodes that include the node in question.

Eigenvector centrality [11] heuristic Eigenvector centrality computes the centrality for a node based on the centrality of its neighbors. A high eigenvector centrality measure means that a node is connected to many nodes who themselves have high measures.

Hybrid centrality [5] heuristic At each step, remove the node with the highest hybrid centrality still available. The hybrid centrality of a node is defined as the product between the Eigenvector centrality of said node and the sum of the coreness, as defined by the K-Shell method, of its neighbors.

Mixed degree decomposition [5] heuristic At each step, randomly remove a node from the M-Shell with the highest coreness. The M-Shell is computed in a similar way to the K-Shell decomposition, but considers the node degree as a sum of the residual degree (number of edges linking the node to available nodes) and the exhausted degree (number of edges linking the node to removed nodes) multiplied by a scalar, λ, which we set to 0.7 as suggested in the original paper [5].

D. Viral Containment Simulation

To accurately test each heuristic, we run multiple iterations of our experiment: first, we order all nodes according to the chosen heuristic; then we remove a determined percentage of these nodes ϵ; and finally we run a SIS simulation with a predetermined rate of initial infection p, a probability of each node infecting its neighbors β and of recovering γ. We standardize all simulations on a γ with value of 40%. The initial rate of infected nodes is set to 5%. We simulate an SIS spreading model, until this model converges into a stable configuration. We consider 8 node removal percentages, from 0.0% up to 87.5%.

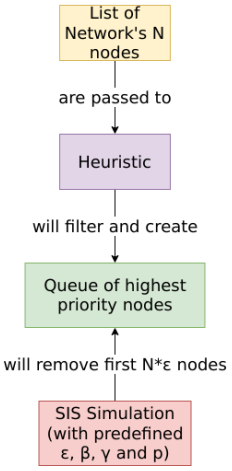


Fig. 2. High level view of the experimental setup. Our simulations are then supplied with the appropriate ϵ, β, γ and p values.

E. Implementation

The entire implementation of the experimental model was setup relies on the graph library *networkx* [12], for Python.

IV. Results and Discussion

In this section we present and discuss the results of our simulations.

Figure 3 shows the results of running the method described in Section III-D with the strategies listed in Section III-C. In each sub figure, a different attack heuristic is displayed. The x axes represent the probability of a susceptible node becoming infected, β, in the SIS model. The y axes represent the proportion of infected nodes at the point of convergence of the SIS model simulation. Each line plot represents an percentage of removed nodes from the network, ϵ. The error bars plot the 95% confidence interval.

Figure 4 shows the same results but for a fixed value of β. It allows for a better comparison between different heuristics.

The x axes represent the proportion of removed nodes, ϵ. The y axes represent the proportion of infected nodes at the point of convergence of the SIS model simulation. Each line plot represents a different attack heuristic. The error bars plot the 95% confidence interval.

Figure 3 (a) shows the baseline case, i.e. the Random heuristic. We observe that the proportion of infected nodes at the convergence point lowers as we remove nodes from the network. This is intuitively explained by the fact that removing nodes decreases the robustness of the network [13], hence limiting the spread.

Figures 3 (b)-(h) show the remaining heuristics' behavior separately. Figure 4 allows for direct comparison between the different heuristics.

We observe that the *Degree Centrality heuristic* is the best-performing attack heuristic, achieving the lowest proportion of infected nodes at all attack intervals, as shown by Figures 4 (a)-(d). Additionally, we observe the high impact of this heuristic on the dynamic of the spread in Figure 3 (f). For the higher percentages of removed nodes (i.e. 75.0%, 87.5%) the proportion of infected nodes is drastically lower when compared to the baseline case.

The *Mixed Degree Decomposition heuristic* ranks second, followed by the *Betweenness Centrality heuristic*. The *K-Core*, *K-Shell*, and *Hybrid Centrality* heuristics all perform similarly, but not very distinctly from the baseline heuristic. The *Eigenvector heuristic*, surprisingly, performs worse than the baseline.

These results highlight that known algorithms capable of identifying influential nodes do not necessarily translate into good one-shot attack heuristics. As Holme et. al [14] point out, attacks on networks change the structure of the network, invalidating the computed values at the beginning of the attack process. Holme et. al also show that if these values are re-computed after each removal process, better attack results are obtained.

Therefore, our results show that, in low-resource environments where there is no capacity for re-computing the given heuristics, using heuristics which highly depend on the structure of the network (e.g. Betweenness Centrality, Eigenvector Centrality) is to be avoided. On the other hand, heuristics which depend on more local information (e.g. Degree Centrality, Mixed Degree Centrality) tend to maintain the performance as the network's structure changes.

V. Future Work

Due to the simplicity of the heuristics chosen, further work could be done on analysing more complex heuristics. One interesting possible angle of research resolves around training Graph Neural Networks (GNN's) [15] in the discussed task, leveraging the structural knowledge of the network. Another angle could expand the presented work to different network models, such as the Erdős–Rényi model [16] or Scale-Free networks [17], and compare the behavior of the heuristics in these different setups. The spreading model can also be changed: instead of only analysis of SIS models, one could analyse other different, more elaborate and real models, such as SIR [8], or even more complex non-linear models [18].

VI. Conclusion

In this paper, we have presented 7 different one-shot heuristics for prioritizing nodes to be attacked in the context of viral containment. We have proposed and executed an experimental setup to test the proposed heuristics against an SIS model running on real-world networks generated through the LFR benchmark. Our results highlight the need for caution when selecting the heuristics to use in one-shot environments, as the existing techniques for identifying influential nodes do not necessarily translate into effective one-shot heuristics. We also suggest that structure oriented one-shot heuristics tend to perform worse than local information oriented ones.

In the future, these results may prove useful in analysing and containing new viral spreads, in large real-world communities.

References

[1] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, Oct 2008. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.78.046110

[2] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," *Nature physics*, vol. 6, no. 11, pp. 888–893, 2010.

[3] L. Jiang, X. Zhao, B. Ge, W. Xiao, and Y. Ruan, "An efficient algorithm for mining a set of influential spreaders in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 516, pp. 58–65, 2019.

[4] C. Li, L. Wang, S. Sun, and C. Xia, "Identification of influential spreaders based on classified neighbors in real-world complex networks," *Applied Mathematics and Computation*, vol. 320, pp. 512–523, 2018.

[5] S. Ahajjam and H. Badir, "Identification of influential spreaders in complex networks using hybridrank algorithm," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[6] L.-l. Ma, C. Ma, H.-F. Zhang, and B.-H. Wang, "Identifying influential spreaders in complex networks based on gravity formula," *Physica A: Statistical Mechanics and its Applications*, vol. 451, pp. 205–212, 2016.

[7] G. F. de Arruda, A. L. Barbieri, P. M. Rodriguez, Y. Moreno, L. da Fontoura Costa, and F. A. Rodrigues, "The role of centrality for the identification of influential spreaders in complex networks," 2014.

[8] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, vol. 115, no. 772, pp. 700–721, 1927.

[9] S. B. Seidman, "Network structure and minimum degree," *Social networks*, vol. 5, no. 3, pp. 269–287, 1983.

[10] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[11] P. Bonacich, "Power and centrality: A family of measures," *American journal of sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

[12] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[13] M. Bellingeri, D. Bevacqua, F. Scotognella, R. Alfieri, Q. Nguyen, D. Montepietra, and D. Cassi, "Link and node removal in real social networks: A review," *Frontiers in Physics*, vol. 8, p. 228, 2020.

[14] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Physical review E*, vol. 65, no. 5, p. 056109, 2002.

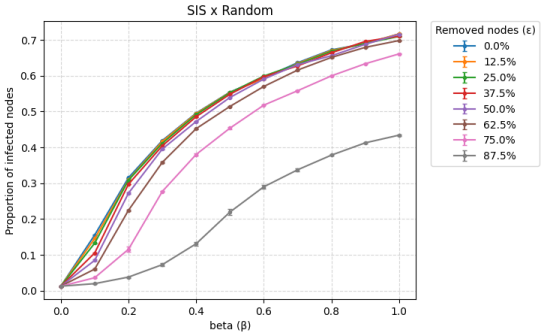
[15] Z. Liu and J. Zhou, "Introduction to graph neural networks," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 2, pp. 1–127, 2020.

[16] P. Erdos, A. Rényi *et al.*, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

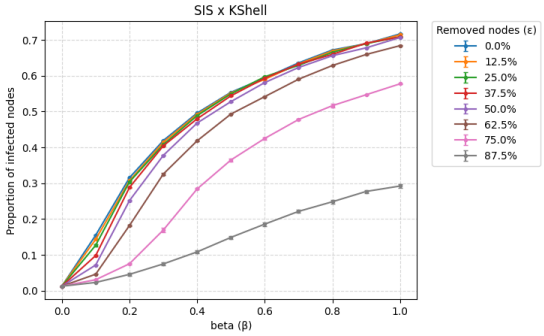
[17] A.-L. Barabási and E. Bonabeau, "Scale-free networks," *Scientific american*, vol. 288, no. 5, pp. 60–69, 2003.

[18] S. Lehmann and Y.-Y. Ahn, *Complex spreading phenomena in social systems*. Springer, 2018.

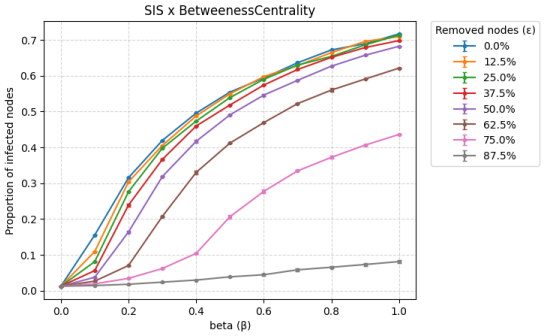
Appendix A - Experimental Results



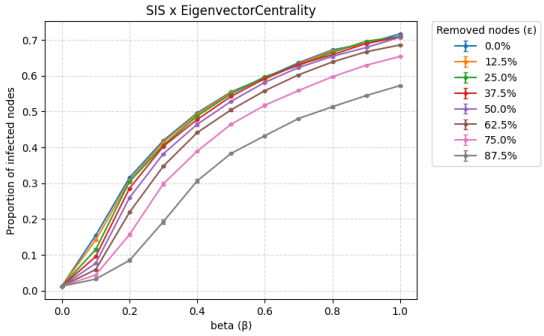
(a) Proportion of infected nodes at the stable setting of the SIS simulation, with the Random heuristic, in function of β.



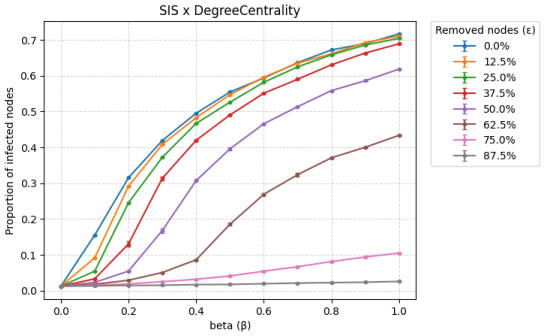
(b) Proportion of infected nodes at the stable setting of the SIS simulation, with the K-Shell heuristic, in function of β.



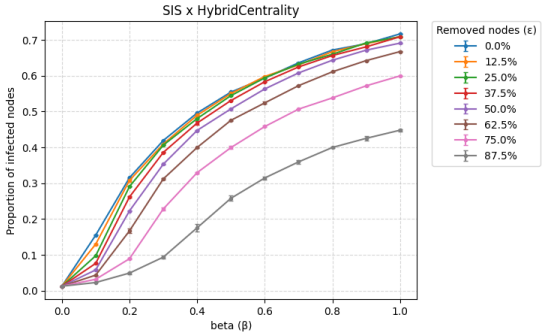
(c) Proportion of infected nodes at the stable setting of the SIS simulation, with the Betweenness Centrality heuristic, in function of β.



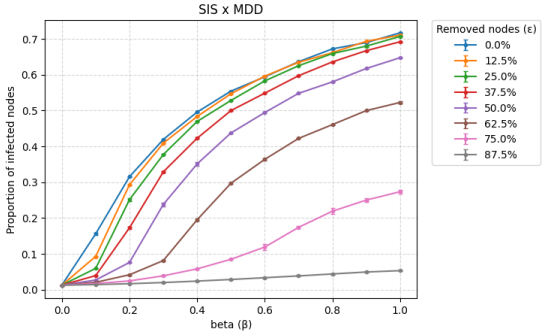
(d) Proportion of infected nodes at the stable setting of the SIS simulation, with the Eigenvector centrality heuristic, in function of β.



(e) Proportion of infected nodes at the stable setting of the SIS simulation, with the Degree Centrality heuristic, in function of β.

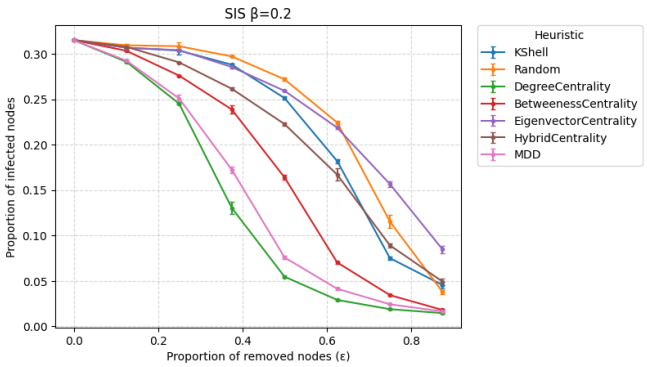


(f) Proportion of infected nodes at the stable setting of the SIS simulation, with the Hybrid Centrality heuristic, in function of β.

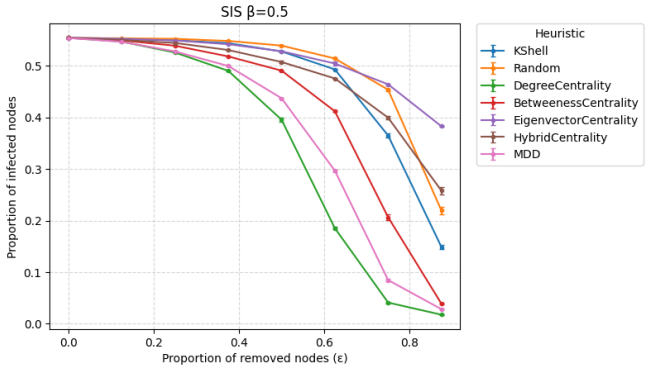


(g) Proportion of infected nodes at the stable setting of the SIS simulation, with the Mixed Degree Decomposition heuristic, in function of β.

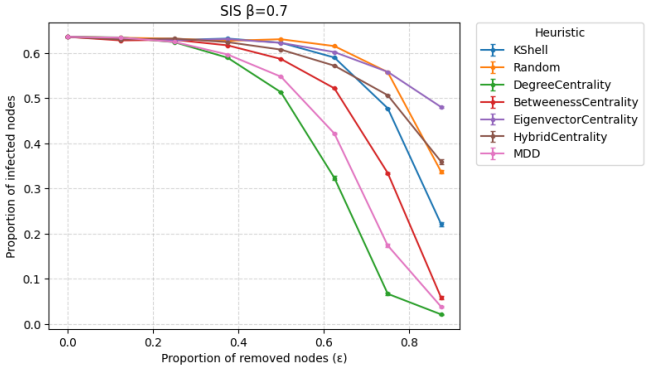
Fig. 3. Results of the simulations of the SIS model for each attack heuristic in function of the probability of an infected node infecting a susceptible one (β).



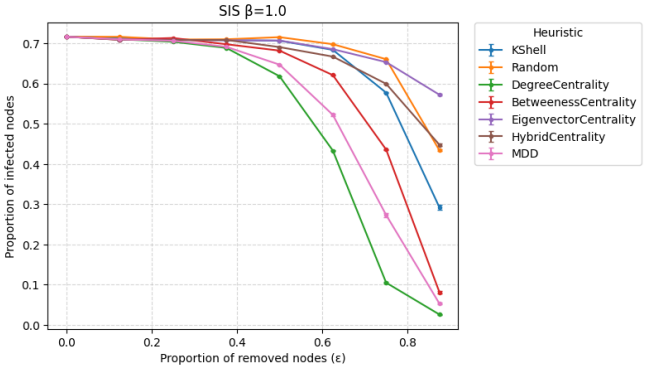
(a) Proportion of infected nodes at the stable setting of the SIS simulation, in function of the proportion of removed nodes ϵ.



(b) Proportion of infected nodes at the stable setting of the SIS simulation, in function of the proportion of removed nodes ϵ.



(c) Proportion of infected nodes at the stable setting of the SIS simulation, in function of the proportion of removed nodes ϵ.



(d) Proportion of infected nodes at the stable setting of the SIS simulation, in function of the proportion of removed nodes ϵ.

Fig. 4. Results of the simulations of the SIS model for four fixed values of the probability of an infected node infecting a susceptible one (β), in function of the proportion of removed nodes (ϵ)