Dynamically Blocking Contagions in Complex Networks by Cutting Vital Connections

Pavlos Basaras[1], Dimitrios Katsaros[1], and Leandros Tassiulas[1,2]

[1]Department of Electrical & Computer Engineering, University of Thessaly, Greece
[2]Department of Electrical Engineering & Yale Institute for Network Science, Yale University
[1,2]{pabasara,dkatsar}@inf.uth.gr leandros.tassiulas@yale.edu

***Abstract*—With the emergence of Online Social Networks (OSNs), as the most popular medium for advertisements, as source of knowledge and information, the emergence of malicious contents (viruses, false rumors, etc..) has become a critical issue that requires immediate attention. In this study we investigate on blocking the contagion of malicious things dynamically, by continuously fighting the diffusion near the source of misinformation under the Susceptible-Infectious-Recovered (SIR) model. We focus on protecting networked populations by removing key connections between nodes, and show via experimental results, that by following the infection the contagion can be controlled more efficiently and even being stopped in the earliest steps. We modify a well studied heuristic from the literature of graphs, and show that our proposed technique significantly outperforms what we believe the state-of-the-art competitors by successfully confronting the infection in real networks.**

I. Introduction

Controlling epidemic outbreaks [1], i.e., the diffusion of "troublesome" contents over the social medium, has received increased attention over the last decade. Most of the so far proposed studies focus on immunization techniques that remove node-users from a network to block the outspread of undesired propagations [2][3][4]. It has been shown that removing the *bridge-nodes* (nodes connected to different communities) or nodes connected to many other nodes (*hubs*), can quite often be an effective solution. However with such methods the immunized entities are completely isolated from the rest of the networked society, while at the same time a network's integrity may be significantly affected. Such drawbacks prompted the research community towards edge-based immunization methodologies for controlling epidemic outbreaks [5][6][7] since the removal of edges is considered as a more realistic approach. For instance removing connections between users e.g., friendships in Facebook, is a more feasible countermeasure than removing individuals from the entire Facebook society. As another example we may consider different type of networks e.g., in military communications shutting down a router may not be an available option.

A similar problem to our case study is the issue of identifying a minimal subset of nodes or link connections between them, whose removal will minimize the number of potential infected nodes. Researchers often nominate greedy algorithms to address the issue or propose approximations on the basis of greedy strategies [8][9]. While the aforementioned studies and the current work, focus in deleting network components (e.g., nodes or edges) to protect a networked environment, other studies apply different policies e.g., by utilizing *protectors* who will disseminate good information to counter the malicious propagation as in [10].

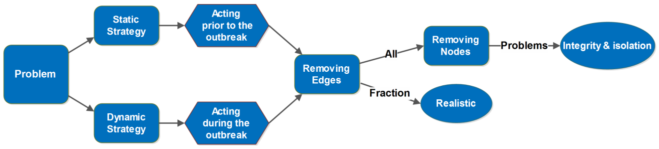

Fig. 1. Generalized framework for blocking epidemic outbreaks in Complex Networks. This article focuses on dynamic strategies and edge removing mechanisms to hinder the spread of misinformation.

Heretofore we may consider node removal approaches as a particular case of edge-based techniques, where the deletion of all connections of a node results in its abscission from the rest of the network. As a next step we group previous works, in terms of how they "protect" a network from false propagations i.e., static or dynamic control strategies. A static control approach vaccinates network components, prior to the outbreak, by selectively removing a limited β number of nodes or connections, based for instance on different centrality measures or path counting approaches. Although we obtained a number of good strategies for priorly dealing with an epidemic, what more can be achieved by dynamically facing the contagion?

In this article we focus on controlling epidemics by dynamically choosing which connections to remove as we closely follow the contagion within the diffusion steps. At each discrete time step a number of β connections may be removed from the network as countermeasures from the authorities. For example consider an event much like KoobFace [11] and a specialized personnel with the knowledge of the currently infected accounts. Instead of taking drastic measures to remove all the connections from the infected users and block the outspread of the virus, the staff could focus on specific interactions among all immediate endangered accounts to hinder or stop the malware from propagating without completely disrupting the networked environment. However we cannot expect for the virus to stay idle while the personnel operates to protect the network and thus we assume that we have a limited number of actions-time before it further propagates.

By mining the knowledge out of a network's current state i.e., origin of infection and susceptible surroundings, a more profound and efficient selection among all possible and proximal edges may be adopted, which intuitively will better hinder the contagion. To the best of our knowledge little work is done in confronting an epidemic dynamically. In [12] the authors proposed a dynamic approach for fighting

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

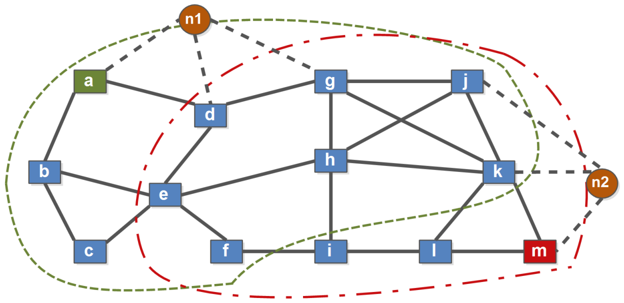

Fig. 2. In the current time step (t) the infected nodes are assumed to be 'a' and 'm' whereas n1 and n2 are the infected sources of the immediate previous step (t-1) which are now immunized (removed). The dashed lines correspond to the three hop abstract network images, as seen from the perspective of the current infected sources.

epidemics, but they focused on strategies for healing already infected nodes under the susceptible infectious susceptible (SIS) model. Here we follow the contagion as it evolves and propagates through node interactions and propose an algorithm that detects critical connections based on their diffusion capabilities namely, *Critical Edge Detector* (CED). These edges will constitute our targets for immunization in our effort to save the largest possible fraction of a complex network when bounded by a limited number of actions-deletions per step. So far the general framework for blocking contagions is shown in Figure 1. Our analysis lies in the lower flow of the diagram.

The rest of the article is organized as follows: in Section II we provide a formal description of the addressed issue. Next in III we detail our proposal. Section IV briefly describes the competing techniques and the evaluation criteria as well as the performance of the competing heuristics. Finally in V the conclusions.

II. Problem Formulation

Let $G(V, E, c_e)$ denote an undirected Complex Network of V nodes connected through E links, where each edge is associated with a positive cost c_e for deletion. The dynamic version of the problem confronts us with the following situation: at each discrete time step t, we have a number of immediate vulnerable nodes which we will try to protect, recovered nodes who were infected in past steps and can no longer be affected by the malicious propagation, and finally the infected ones who will now try to infect their susceptible neighbors. To simulate the diffusion process of undesired data over G, we utilize the susceptible-infectious-recovered model (SIR) which unfolds in discrete steps. Nodes who are infected during the dissemination process are considered as the lost fraction of nodes. Given a budget β of available deletions per step $-$equal cost for the removal of any edge$-$ we search for those connection whose deletion will result in the least number of lost nodes at the end of the malicious propagation. As a next and final constraint we consider that the "authorities" exhaust all their available resources at each time step i.e., resources cannot be saved for later use.

III. Critical Edge Detector (CED)

For our method we focus on the infected nodes of each step, to create the *Infected-Source-Networks* ($ISNs$) emanating from each individual 'tainted node' x at time step t. The $ISNs$ are created from the susceptible nodes within the n-hop neighborhood of each infected source x (including x) and the link-connections between them namely, ISN^n_x. Our work is limited in short distances from the originators in order to fight the contagion near the source of the problem, and inhibit its transition as much as possible. An illustrative example is given in Figure 2. Initially we assume that the infection came from nodes n1 and n2 at time t-1 who successfully infected nodes a and m. The 3-hop infected source networks emanating from the current infected nodes at time t, ISN^3_a and ISN^3_m, are shown with green and red dashed lines respectively. Note that the infected sources n1 and n2 from the previous step (t-1) are excluded from our selection in all subsequent steps, since they can no longer contribute in the propagation as the diffusion model implies (c.f. IV-B2).

To quantify the importance of an edge (i, j) in an ISN^n_x, we calculate the number of shortest paths (using Dijkstra's algorithm) emanating from the infected source x to all other nodes in the current ISN^n_x that (i, j) appears, with respect to the total number of those paths as follows:

$$ISN^n_x(i,j)_t = \frac{sp^n_{ij}(t)}{sp^n_t} \tag{1}$$

$sp^n_{ij}(t)$ is the number of shortest paths that the edge (i, j) appears at t step emanating from x, and sp^n_t stands for the total number of those paths.

The concept of *Single Source Shortest Path* (SSSP) is a widely used method in the science of networks, well suited for the facet we are addressing in the present study, as we dynamically deal with a contagion directly at its source to block the outspread. At this point we should note that by grounding the source of the infection i.e., pinpointing the malicious sources, we understand the direction of the propagation. Our proposed technique uses the course-evolution of the diffusion (towards the susceptible environs) to its advantage, and locate those links which will hinder the malicious act to the largest possible extent. However, not all ISNs are of equal size i.e., in the number of susceptible nodes or connections. In fact this is a varying parameter that must be taken into consideration, since edges located in relatively sparse ISNs, may well be overestimated for their spreading potential. Thus we need to include a notion of density for the *end-point* node. Since we noted the course of a virus, the end-point node is a potential direction, e.g., in Figure 2, k is the ending node of m-k. The density for the end-point-node j is measured by the formula:

$$d_j = s_j - P_j + \sum_r (s_r - P_r - M_{rj}) \tag{2}$$

where s_j is the number of currently susceptible neighbors of j, P_j is the fraction of nodes out of s_j with at least one infected neighbor, r are the susceptible neighbors of j and M_{rj} denotes the common neighbors between r and j. If j leads to a dense region of susceptible neighbors the importance of the connection will be boosted accordingly, whereas for sparse vicinities d_j will be lower.

Finally the final index for each edge as accumulated by CED is given by the formula:

$$CED(i,j) = ISN^n_x(i,j) \cdot d_j \tag{3}$$

The pseudo-code for our technique is given in Algorithm 1. Henceforth we assume that the infected source networks are obtained from the 2-hop neighborhood of the originator i.e., ISN_x^2.

Algorithm 1 Dynamic strategy for deleting edges

```
procedure EDGE RANKING ($CED$)
    for each node $x$ in Infected State at time $t$ do
        Create the corresponding $ISN_x^n$s at $n$ distance
        Use Dijkstra algorithm to obtain shortest paths
        Rate links based on their occurrence
    end for
    Calculate $d_j$ to obtain the density of end-point node $j$
    Calculate $CED(i,j)$ to obtain the final index of $(i,j)$
end procedure
```

IV. Performance Evaluation

A. Datasets

A summary for the base attributes of the experimented networks is listed in Table I. α stands for the epidemic threshold of transmissibility calculated for each respective network [13], and k-core illustrates the greatest shells (the core of a network) as identified by the k-shell decomposition algorithm [14]. The experimental networks were selected based on the density (connections) of their core to evaluate the competitors in diverse networked environments; *Hamsterster*: a social network, *Pretty Good Privacy* (PGP): secure information interchange network, *Oregon-2*: an autonomous system graph from May 26 2001, and finally an email contact network *Enron*. For more details on the datasets please refer to http://konect.uni-koblenz.de/ and http://snap.stanford.edu/data/index.html.

TABLE I. Network Base Attributes

Network	No. of Nodes	No. of Links	k-core	$\alpha(\%)$	Type
Hamsterster	2,426	16,631	24	2.5	Social
PGP	10,681	24,316	31	5.5	Contact
Oregon-2	11,461	32,730	31	5.5	AS
Enron	36,692	367,662	43	1.5	Email

B. Experimental Design

1) Initiating the Cascade: The origin of the infection i.e., the initially infected nodes, is an important feature that affects the diffusion dynamics. For instance, if the originators are within a sparsely connected neighborhood, even with a limited number of available deletion per step, the diffusion is very likely to be inhibited. Similar performance will be achieved, if the origin of the infection is placed in the periphery of a network as identified by the k-core algorithm. Such configurations are trivial for our experimentation. On the contrary, if the originators are nodes in denser regions of a network, successfully inhibiting the outspread of undesired data will prove to be a more challenging task.

To this end, in a similar approach to [6], we initiate the infection from the top-10 most connected nodes (*hubs*) within the highest k-cores of each network. It is safe to assume that initiating the infection from hub-core nodes is no trivial task −maybe the worst case scenario− since the core represents well connected node-users who are "buried" deep within the network structure.

2) Propagation Model: For the diffusion model as noted in [15], the SIS (like flu) suggests no immunity for the interacting nodes, whereas the SIR offers permanent immunization (like mumps). Here we study the penetration of a virus in a networked environment, where immunized nodes cannot be reinfected and thus focus on SIR which unfolds in discrete steps:

- **S** state where a node is vulnerable to infection.
- **I** state where an infected node will try to infect its susceptible neighbors and succeed with probability λ.
- **R** state where infected nodes recover and cannot be reinfected.

In the initial phase all nodes are in the susceptible state, except the initially selected nodes in I. Generally, an infected node at time step t has a single chance to infect its susceptible neighbors and succeeds with probability λ. Immediately after the node enters the R state at t+1, and can no longer be infected in subsequent steps. The process ends when there is no node left in I.

3) Removing Connections: In this study, we follow the diffusion dynamically i.e., as it unfolds through node interactions, and thus the links that constitute all possible options for removal at each time step, are those in direct contact with the infected sources. As far as the constraint for removing edges per step is concerned, we take 1% of the total connections of each network and name this fraction of edges as *thres*. The x-axis in each plot represents the percent out of *thres* cut in each diffusion step, namely β number of edges. We limit our experiments to small β values per step, to evaluate the competitors ability in detecting the most efficient interactions for blocking malicious diffusions.

4) How to evaluate the performance: In order to obtain unbiased results, for each method we repeated over 1000 diffusion processes. The error-bars in each plot represent the confidence for the interval of the mean, i.e., over the iteration that we repeated the SIR process the true average value is bounded within the specified range. The probability of diffusion among interactions (λ), is chosen based on the epidemic threshold α of each respective network. However in $Hamstester$ due to its lower connectivity we had to use a relatively higher value to obtain significant results.

The impact of each method is measured based on the fraction of the network affected by the false rumor-virus at the end of the SIR process, i.e., number of nodes in R state (lost nodes). The evaluation is carried out in two distinct SIR processes. In the first, we measure the fraction of lost nodes when no protection algorithm is applied, and in the second the diffusion is re-initiated to utilize the competing techniques.

C. Competing Methods

The presentation of the addressed issue in this work is original and thus the selection of appropriate competing techniques is crucial. Here we list our selection in the competing methods and also exclude those that can not be applied. Note that only techniques for undirected networks are included. (*i*) highly connected nodes are noted by many studies as influential spreaders, and thus in [7] the strength of connections is measured by the product of the degrees of the nodes incident upon a link ($aDegree$). Note that for this approach only susceptible neighbors frame the degree of a node, since these are the nodes we will be trying to protect. For the current competitor the edges are selected in decreasing order of $aDegree$ until β is reached at each time step.

Ranking the importance of edges e.g., by using centrality metrics, is not the only means to select a limited number of interactions. In [6] the authors use a different approach

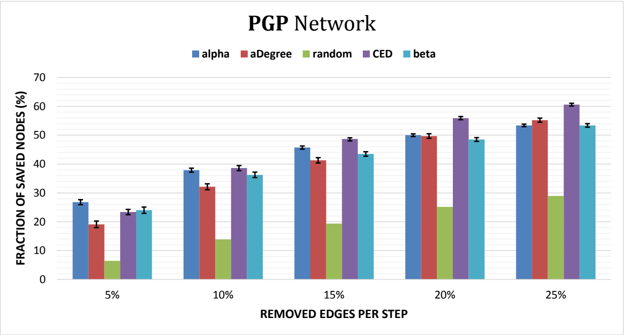

Fig. 3. The strength of the propagation is 6%. The initially infected set is connected to the immediate vicinity with 548 connections whereas the lost fraction of nodes for the unblocked diffusion is about 280 nodes. As we increase in the x-axis CED's better performance becomes more evident.

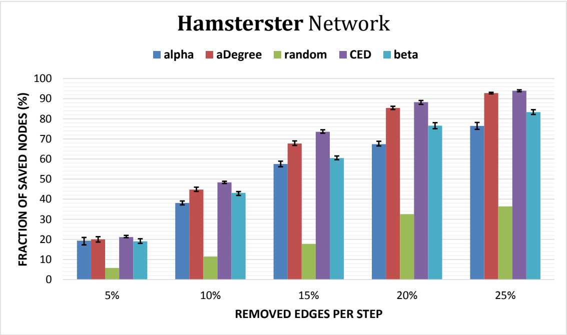

Fig. 4. The strength of the propagation is 4%. The initially infected set is connected to the immediate vicinity with 410 connections whereas the lost fraction of nodes for the unblocked diffusion is about 360 nodes. For this weakly connected network all methods illustrate a good performance.

by strategically selecting which edges to remove, in order to decrease the probability of cascade and salvage the largest number of potential nodes. In a similar approach −although under a stochastic and different diffusion model− we construct a strategic deletion of edges to secure the largest number of immediate and endangered individuals.

(*ii*) For the first strategy, immediate susceptible nodes are ranked based on their vulnerability i.e., number of neighbors in I state. Individuals with the least number of infected neighbors are firstly treated and so on until the available budget for this step is exhausted. In order to avoid consuming a significant amount of resources to save a single individual, for nodes with vulnerability greater than one we remove one connection at a time. If one edge is removed from all vulnerable nodes and there are still available resources we re-initiate the procedure until β is consumed. Nodes with the same vulnerability are ranked in decreasing order of their susceptible degree. Note that nodes with only one infected neighbor are completely protected in this round. We name this strategy *alpha* where we try to decrease the probability of a cascade throughout the diffusion steps.

(*iii*) As a second strategy, namely *beta*, we rank all susceptible nodes in direct contact with one or more infected sources in decreasing order of their susceptible degree i.e., number of still unaffected neighbors. With this strategy we try to reduce the number of interaction that lead to the highly connected individuals in each step. Likewise in *alpha* we don't want to consume most of our resources for protecting specific individuals and thus we follow the same policy. Note that when the budget β for deleting edges is sufficient to remove the same number of connections from all immediate susceptible nodes (rare occasion), it applies that $alpha \equiv beta$.

(*iv*) Finally a random selection of edges (*random*) is used as a baseline to create a lower bound of performance. Here a uniform selection among all possible links is applied.

In [5] the K-$EdgeDeletion$ technique was proposed based on the leading eigenvalue of a network's adjacency matrix. Here the strength of a link (i, j) is determined by the product of the leading left and right eigenvectors $u(i) \cdot v(j)$ respectively. Although it was proven an effective static strategy for blocking epidemic outbreaks, to our understanding the eigendecomposition of an adjacency matrix cannot be applied in our case study for the following reasons. Due to the dynamic nature of the addressed issue, in order to apply the aforesaid technique at each time step, we must *either* use the entire remaining network −excluding nodes in R state− and proceed as the algorithm unfolds, *or* create a connected component which would include all infected sources and susceptible surroundings within a certain hop distance, and finally focus in the obtained ranking for those links which are directly connected to an infected source.

Although the second approach may be applicable in the first steps of the SIR when the initially infected nodes may be in close distance, in the later steps were the infected sources become distant to each other we approximately fall in the first case scenario. Nonetheless applying such adjustments to the original algorithm will distance our work from its dynamic nature and thus we do not include K-$EdgeDeletion$ in our evaluation.

D. Results

1) Increasing in the number of deletions per step: As a first step to our evaluation we illustrate the results from Figures 3 to 6. The y-axis represents the fraction of saved nodes i.e., the percent of node-entities that each respective method managed to secure, with respect to the unblocked outcome of the propagation. It can be seen that the proposed identification technique performs extensively well in most of the observed cases. Our results indicate that cutting of edges within certain limited regions of a network (the $ISNs$) which reside in many shortest paths, is the most effective solution for blocking or hindering the infection dynamically.

To better analyze the performance of the competitors, let us consider the evaluated networks with respect to the connectivity of their initially infected core. The selection of the initial infected seed set out of the most connected nodes within the core shell of PGP and $Hamstester$, form a weakly connected set with average degree of 54.8 and 41.1 respectively. It is reasonable to assume for such cases that by blocking the diffusion directly at its source, a relatively good performance would be achieved by all methods. The results illustrated in Figures 3 and 4 confirm this hypothesis. In PGP CED's better performance becomes more evident as we increase in β. When considering $Hamstester$ we focus on the lower values to conclude on the competing algorithms since for the highest steps −due to its low connectivity− all methods manage to significantly block the contamination.

For the $Enron$ and $Oregon$-2 networks in Figures 5 and 6,

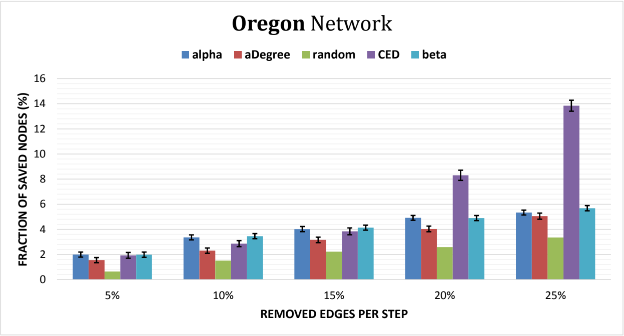

Fig. 5. The strength of the propagation is 6%. The initially infected set is connected to the immediate vicinity with 3400 connections whereas the lost fraction of nodes for the unblocked diffusion is about 1270 nodes. Only the proposed technique manages to hinder the propagation sufficiently in the later steps of β.

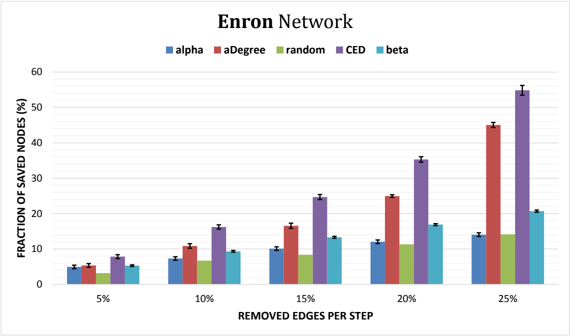

Fig. 6. The strength of the propagation is 2%. The initially infected set is connected to the immediate vicinity with 11285 connections whereas the lost fraction of nodes for the unblocked diffusion is about 2080 nodes. Again the network is better protected by CED.

we analyze a more ambitious case i.e., the average connectivity of the initially infected nodes is 340 and 1128.5 respectively. In these scenarios we expect a more challenging behavior. Indeed, as illustrated, the fraction of saved individuals is significantly less from the previous network cases. For the lower values of β: 5,10 and 15% it appears that none of the evaluated techniques is able to block the contamination significantly i.e. the saved individuals are less than 5% in $Oregon$-2. Only CED manages to save up to about 14% from the lost individuals when β equals to 25%, while the rest of the evaluated techniques illustrate similar behavior with blocking score less than 6%.

Similar results are also reported for the $Enron$ network only here the competitors illustrate a better performance due to the virulence of the propagation set at 2% and the significantly higher number of available deletions per step. For both $alpha$ and $beta$ we observe little improvement in the saved individuals as we increase in β when compared to CED's. To our interpretation, although both strategies performed relatively well for the rest of the experimented networks, it seems that trying to reduce the probability of a cascade (by decreasing the overall connectivity that lead to infected sources or to the most susceptible nodes) as the strategies imply, is not efficient when applied in the core of a well connected network as in this particular case.

Overall we attribute CED's better performance to the following remarks. First, although there are occasions where the contagion cannot be completely stopped in the early steps (due to the infection being rooted deep within a well connected network), by removing the edges as identified by our approach we force the malicious propagation towards longer interacting paths. Thereby more resources can be used in the next steps to inhibit its transition and stop its outspread to more distant regions of a network. Second, by measuring the density of the surroundings of the end-point node, we alleviate traditional drawbacks of shortest path algorithms, since our method will discount the significance of otherwise important links which lead to sparsely connected parts of a network.

2) Increasing in the virulence of the malicious propagation: In this set of experiments, Figures 7 and 8, we evaluate the performance of the competitive methods as we increase the strength of the malicious propagation i.e., increase in λ. The experimented λ values are chosen near the epidemic threshold. We illustrate the results of PGP and $Oregon$-2 networks as we categorized them depending on the average connectivity of their initially infected set from the core. Similar qualitative results were obtained from the rest of the evaluated networks. Finally, the largest β value from our experimentation is selected since increasing in the strength of the propagation will only make the situation more difficult.

In order to measure the influence capability of nodes in complex networks, a problem formally known as detecting influential spreaders in complex network structures [16][17], the virulence of the diffusion should be kept in relatively low values. This is due to the fact that for larger infection values, an epidemic occurs regardless of the characteristics of the node elected as the origin of the infection. In this study, where we initiate the infection from multiple sources from the most connected nodes of the core of each respective network, we expect that blocking the malicious propagation as λ increases will become a very challenging task.

The results in Figure 7 indicate, that when the network is sparsely connected, the infection can still be significantly mitigated even when the virulence of the diffusion is higher —but still near— the epidemic threshold. For the lower λ values $aDegree$ and CED illustrate similar performance, however as we increase in λ the proposed technique significantly outperforms all competing methods. $aDegree$ is affected by the increase of λ around 6% and henceforth its performance starts to decent, whereas reducing the probability of the cascade with both $alpha$ and $beta$ strategies, seems to have an increasing performance that surpasses $aDegree$ when above the epidemic threshold. Nonetheless further increasing in λ will only decrease the fraction of saved nodes that each respective method manages to secure.

By following the performance of the competitors in $Oregon$-2 we observe a different outcome. For this scenario all methods illustrate a decreasing performance in the saved individuals as we advance in λ. However the competitors fail to protect an adequate fraction of the network even bellow the epidemic threshold. Only the proposed technique bears more resistance to the virulence of the propagation and is able to save a significantly larger number of endangered nodes. To our understanding when λ increases beyond a certain threshold —different for each network depending on its connectivity— the diffusion cannot be significantly hindered. This is due to the fact that even by deleting a large number of immediate

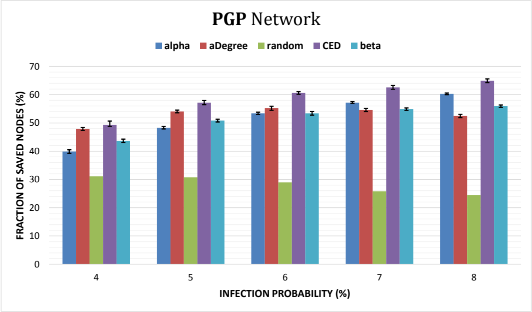

Fig. 7. The y-axis represents the fraction of saved nodes with regard to the lost nodes of the unblocked diffusion (113, 190, 280, 385, 511) respectively. CED illustrates better results by securing a significantly larger part of the network's interacting nodes for all λ values.

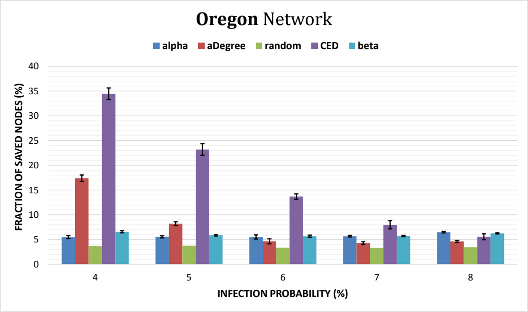

Fig. 8. The y-axis represents the fraction of saved nodes with regard to the lost nodes of the unblocked diffusion (814, 1048, 1270, 1488, 1714) respectively. Our approach seems to be affected by the increase of λ significantly later than its competitors.

connections i.e., increase in β, and thus significantly diminish the available paths from infected nodes to susceptible individuals, when we are bound to the higher values of λ the virus is very likely to survive even within the now few remaining interaction.

V. Conclusion

In this study we take a first step in confronting the diffusion of malicious contents over networked populations dynamically, while we follow the virus as it progresses through node communications. Most of the so far proposed techniques focus on static strategies, however we believe that the problem is dynamic in nature and must be addressed appropriately. We proposed an algorithm that utilizes well studied heuristics from the literature of graphs, which was found to be quite effective in blocking the outspread of the diffusion. We used a number of representative competitive methods and strategies −what we believe baseline approaches for the dynamic facet of the addressed problem− to evaluate the impact of our method. Our technique was found to be more efficient by securing the largest fraction of individuals almost in all the observed scenarios. Finally we conclude that when increasing the strength of the prorogation above the epidemic threshold, successfully hindering the propagation can be a very challenging task in a well connected network. Nonetheless CED illustated a more resistant behavior in the increase of the virulence of the propagation.

Acknowledgments: This research has been partially funded by the EU Project 318514 "Convergence of wireless Optical Network and IT rEsourcesiN support of cloud services" (CONTENT). D. Katsaros has been co-financed by the EU (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales - Investing in knowledge society through the European Social Fund in the context of the project "Social network aware cognitive radio networks – SoCoNet (no. 85459)".

References

[1] S. Eubank, V. S. Anil-Kumar, M. Marathe, A. Srinivasan, and N. Wang, "Structure of social contact networks and their impact on epidemics," in *AMS-DIMACS Special Issue on Epidemiology*, 2006, pp. 181–213.

[2] L. Hebert-Dufresne, A. Allard, J. G. Young, and L. J. Dube, "Global efficiency of local immunization on complex networks," *Nature Scientific Reports*, vol. 3, 2013.

[3] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, and Y. Bi, "Least cost rumor blocking in social networks," in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2013, pp. 540–549.

[4] K. Gong, M. Tang, P. M. Hui, H. F. Zhang, D. Younghae, and Y. C. Lai, "An efficient immunization strategy for community networks," *PLoS ONE*, vol. 8, no. 12, 2013.

[5] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation," in *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2012, pp. 245–254.

[6] C. J. Kuhlman, G. Tuli, S. Swarup, M. V. Marathe, and S. S. Ravi, "Blocking simple and complex contagion by edge removal," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2013, pp. 339–408.

[7] Z. H.-F., K.-Z. Li, X.-C. Fu, and B.-H. Wang, "An efficient control strategy of epidemic spreading on scale-free networks," *Chinese Physics Letters*, vol. 26, no. 6, 2009.

[8] M. Kimura, K. Saito, and H. Motoda, "Minimizing the spread of contamination by blocking links in a network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 2, 2008, pp. 1175–1180.

[9] A. Goyal, F. Bonchi, L. V. S. Lakshmanan, and S. Venkatasubramanian, "On minimizing budget and time in influence propagation over social networks," *Social Network Analysis and Mining*, vol. 3, no. 2, pp. 197–192, 2013.

[10] N. P. Nguyen, G. Yan, and M. T. Thai, "Analysis of misinformation containment in online social networks," *Computer Networks*, vol. 57, no. 10, pp. 2133–2146, 2013.

[11] K. Thomas and D. M. Nicol, "The Koobface botnet and the rise of social malware," in *Proceedings of the International Conference on Malicious and Unwanted Software (MALWARE)*, 2010, pp. 63–70.

[12] K. Scaman, A. Kalogeratos, and N. Vayatis, "Dynamic treatment allocation for epidemic control in arbitrary networks," in *Proceedings of the ACM WSDM Workshop on the Diffusion Networks and Cascade Analytics (DiffNet)*, 2014.

[13] C. Castellano and R. Pastor-Satorras, "Thresholds for epidemic spreading in networks," *Physical Review Letters*, vol. 105, no. 218701-1–218701-4, 2010.

[14] 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, pp. 888–893, 2010.

[15] B. A. Prakash, D. Chakrabarti, N. C. Valler, M. Faloutos, and C. Faloutos, "Threshold conditions for arbitrary cascade models on arbitrary networks," *Knowledge and Information Systems*, vol. 33, no. 3, pp. 549–575, 2012.

[16] P. Basaras, D. Katsaros, and L. Tassiulas, "Detecting influential spreaders in complex, dynamic networks," *IEEE Computer magazine*, vol. 46, no. 4, pp. 26–31, 2013.

[17] J. Bae and S. Kim, "Identifying and ranking influential spreaders in complex networks by neighborhood coreness," *Physica A: Statistical Mechanics and its Applications*, vol. 395, no. 1, pp. 549–599, 2014.