**SCIENCE CHINA**
Information Sciences

• **Supplementary File** •

# An Eigenvalue-based Immunization Scheme for Node Attacks in Networks with Uncertainty

Yizhi REN[1], Mengjin JIANG[2], Ting WU[1], Ye YAO[1], Kim-Kwang Raymond CHOO[3] & Zhen WANG[1*]

[1]*School of Cyberspace, Hangzhou Dianzi University* Hangzhou 310018, *China;*
[2]*School of Computer Science and Technology, City* Hangzhou 310018, *China;*
[3]*Department of Information Systems and Cyber Security, The University of Texas at San Antonio* San Antonio, TX 78249, *USA*

## Appendix A Comparison results of proposed algorithms

In this section, we will show the effectiveness of our method in different networks. All these networks have uncertainty edges, and the probability of each edge is generated randomly. The networks we used in our simulation are listed in Table A2.

The benchmark solutions are compared with *NADR* algorithm.

- *Most Probable* (MP). A representative instacne corresponds to a graph containing all edges with probability $p_e \geqslant 0.5$.
- *Approximate B-Matching* (ABM) and *Average Degree Rewiring* (ADR). Two methods are used to extract representative instance and capture the underlying properties of probabilistic networks well [1, 2].

In the following simulation, we will compare the performance of our proposed algorithms (i.e.,*graphCom* and *deepGreedy*) with *Betweenness Centrality algorithm* (`betweenness` for short), *Degree Centrality algorithm* (`degree` for short) and *Netshield algorithm* (`netshield` for short). The description of these three algorithms are list in Table A1.

**Table A1** Algorithms

| Algorithm Name | Definition |
|---|---|
| `betweenness` | immunize the node with the highest betweenness centrality value |
| `degree` | immunize the node with the highest degree |
| `netshield` | immunize the node with the highest *Score* in [3] |
| `graphCom` | immunize the node with the highest *RS* |
| `deepGreedy` | immunize the node with the maximum drop in eigenvalue |

To evaluate the feasibility and effectiveness of our approach, we verified it from simulation network to actual network, from small network to large network. Since the epidemic threshold in a deterministic graph satisfies $\tau = \frac{1}{\lambda}$. Here, let $s$ be the normalized virus strength, we have $s = \lambda\frac{\beta}{\sigma}$ where $\beta$ and $\sigma$ are the infection rate and the recovery rate, respectively.

To evaluate the feasibility and effectiveness of our approach, we verified it from simulation network to actual network, from small network to large network. Since the epidemic threshold in a deterministic graph satisfies $\tau = \frac{1}{\lambda}$. Here, let $s$ be the normalized virus strength, we have $s = \lambda\frac{\beta}{\sigma}$. Here $\beta$ is the infection rate and $\sigma$ is the recovery rate. The number of sample networks M=10000 for $EE$, and the time step T=2000 applies to all networks and $\omega$=5 are used in deepGreedy. Since the sampled networks are all deterministic networks, the largest eigenvalue of each sample network can be calculated by using the common method of solving eigenvalues with adjacency matrix.

Figure A1 illustrates the boxplots for the absolute discrepancy distribution. The red box represents the average discrepancy of per node and the black line represents the standard deviation. For instance, in karate for the representative produced by NADR, the discrepancy between all nodes is in the range [0,0.6]. The average discrepancy is 0.265, and the total discrepancy is 10. Compared with other methods, the total difference value of NADR is much smaller, and the specific data are shown in Table A3.

**Table A2** Network datasets

| graphs | nodes | links |
|---|---|---|
| ER | 100 | 200 |
| USAir | 332 | 2126 |
| karate | 34 | 78 |
| Yeast | 2375 | 11693 |
| football | 115 | 613 |
| Router | 5022 | 6258 |

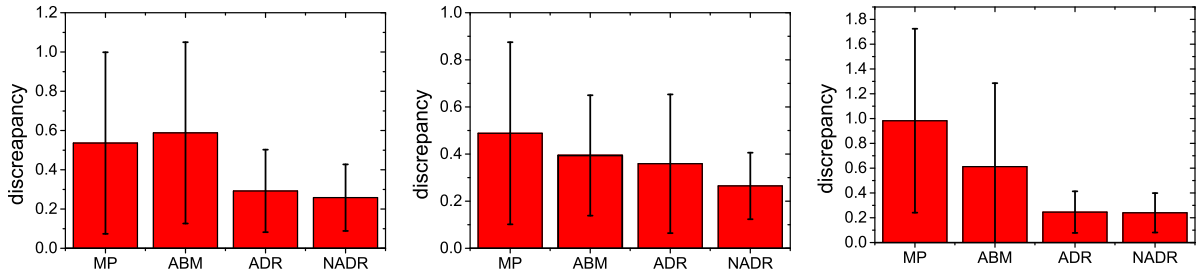


**Figure A1** Variance diagram of the distribution of discrepancy per node.

**Table A3** The distribution of discrepancy in karate

| methods | min(dis) | max(dis) | ave(dis) | total |
|---|---|---|---|---|
| MP | 0 | 2.9 | 0.488 | 16.6 |
| ADR | 0 | 1.5 | 0.359 | 12.19 |
| NADR | 0 | 0.6 | 0.265 | 10 |

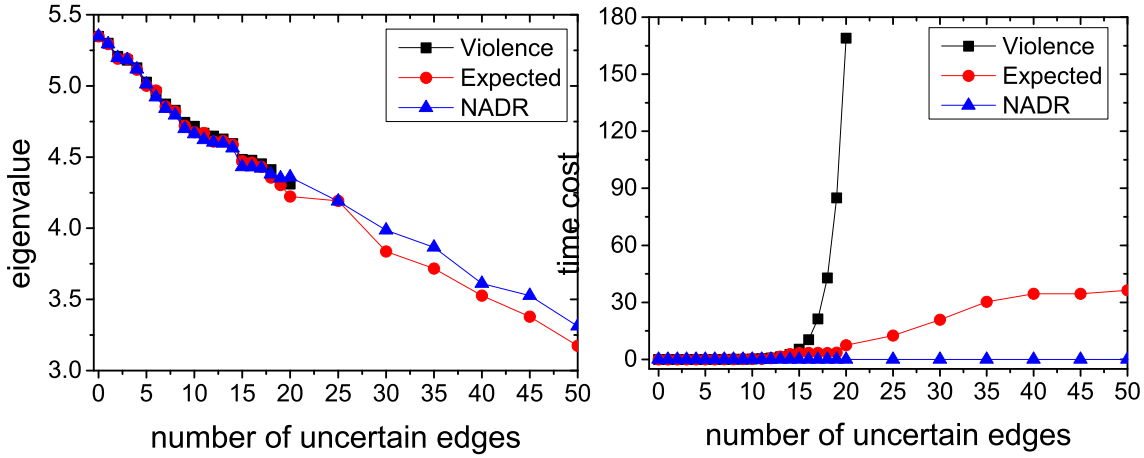


**Figure A2** The variation curves of eigenvalue and time cost with the number of uncertain edges under different methods.

In order to verify the feasibility of extracting a representative instance, we compare the eigenvalues and time cost on the probabilistic graph with different uncertain edges.

Figure A2 shows the results of *number of uncertain edges* vs. *eigenvalue* and *number of uncertain edges* vs. *time cost*. We compare three methods, where *Violence* refers to enumerating all sample networks, *Expected* refers to selecting an appropriate sample and *NADR* refers to extracting a representative instance. When the number of uncertain edges increases, the cost of Violence is very expensive. Therefore, we will not continue to calculate the part with the number of uncertain edges over 20. In Figure A2 (a), we can find that the results are more consistent in the part where the number of uncertain edges is less than 20. For another part, there are some differences in the results of the remaining two methods, but the overall effect is not significant. In Figure A2 (b), as we can seen, the cost of time increases exponentially with *Violence*. The time spent using *Expected* depends entirely on the number of samples selected. When a representative instance is selected using NADR algorithm, the time cost is almost zero.

* Corresponding author (email: wangzhen@hdu.edu.cn)

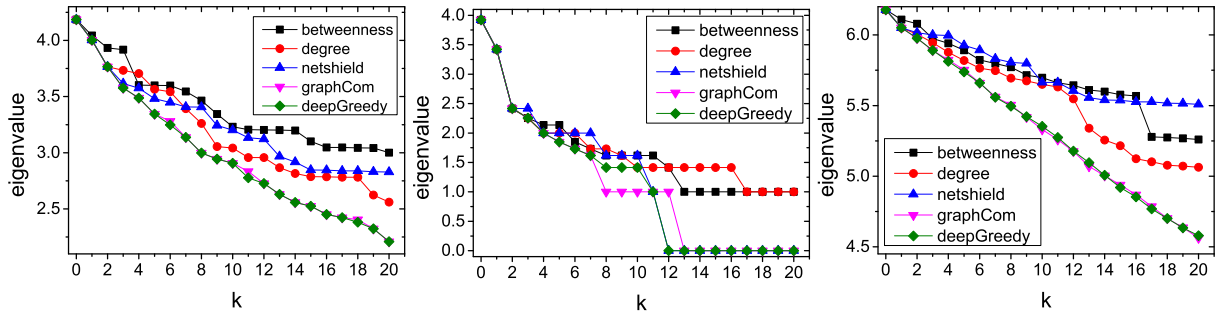


**Figure A3** The largest eigenvalue vs. the number of immune nodes.

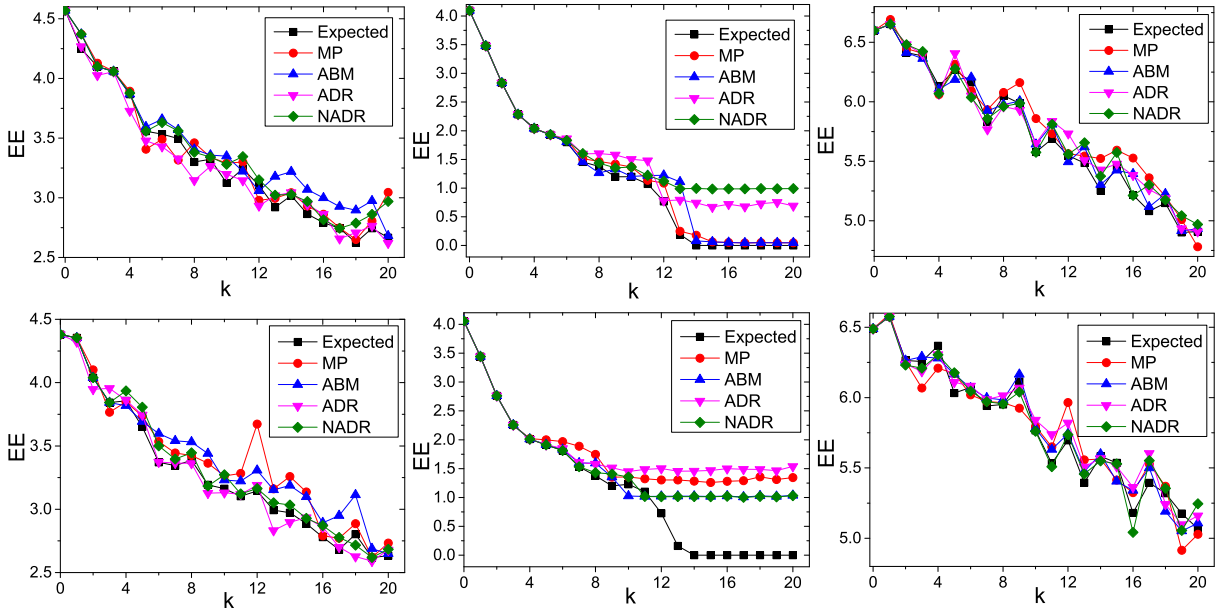


**Figure A4** Largest eigenvalue vs. the number of immune nodes $k$ with graphCom in (a)-(c), with deepGreedy in (d)-(f).

Figure A3 shows *largest eigenvalue* on different number of immune nodes $k$. In different networks, we first select *representative instance* $G^*$ according to *NADR* algorithm and compare our immune algorithm *graphCom* and *deepGreedy* with other methods based on $G^*$. Both *graphCom* and *deepGreedy* are more effective than others. For the same $k$, our algorithm can make the eigenvalue fall faster. And for the same eigenvalue, our algorithm can pick a smaller $k$.

When we use *graphCom* to select immune nodes, different methods to select $G^*$ will affect the accuracy of our choice. The results are shown in Figure A4 (a)-(c). With the change of $k$, the results of most methods are consistent with the EE. In Figure A4 (d)-(f), all the results are derived from using *deepGreedy* to select immune nodes. Except for MP, the results obtained by other methods are relatively stable, where NADR are the best.

We simulate our strategies on uncertainty networks under SIS model. In simulations, we initialize all nodes to be infected. During the propagation process, each node is either as a *Susceptible*(S) state or an *Infective*(I) state. In Figure A5, we show the fraction for infected nodes ($FI$) over time. We try to minimize the value of $FI$ after immunizing some nodes, which means the disease is better controlled. When the virus strength $s \leqslant 1$, the infection will finally die out as shown in (a)-(c). In (d)-(f), the virus strength $s = 1.2$, the infection will survive and become an epidemic. However, since we select $k$ nodes to immune, the largest eigenvalue is reduced. So the $s$ will become smaller (i.e. $s = \lambda\beta/\sigma$). The infection is die out in (d)-(e) and survive in (f). The simulation results in *graphCom* and *deepGreedy* are highly similar, and both are better than other centrality measurement based algorithms. And *graphCom* appears to have a slight advantage over *deepGreedy*.

In a probabilistic graph, the probability of each sample network $Pr(\tilde{G}_i)$ are determined by the number of edges. However, when the number of uncertain edges exceeds 2000, the result will default to 0 based on the requirement of calculation accuracy. Therefore, for larger networks, sampling to compute EE is not feasible. It is sensible to select representative instance to remove uncertainty in place of sampling.

In Figure A6 and Figure A7, we demonstrate the effectiveness of immune strategies in representive instances in large networks. As we can see, the results of our proposed algorithm *graphCom* and *deepGreedy* are relatively consistent, and both of them are superior the other immunization strategies.

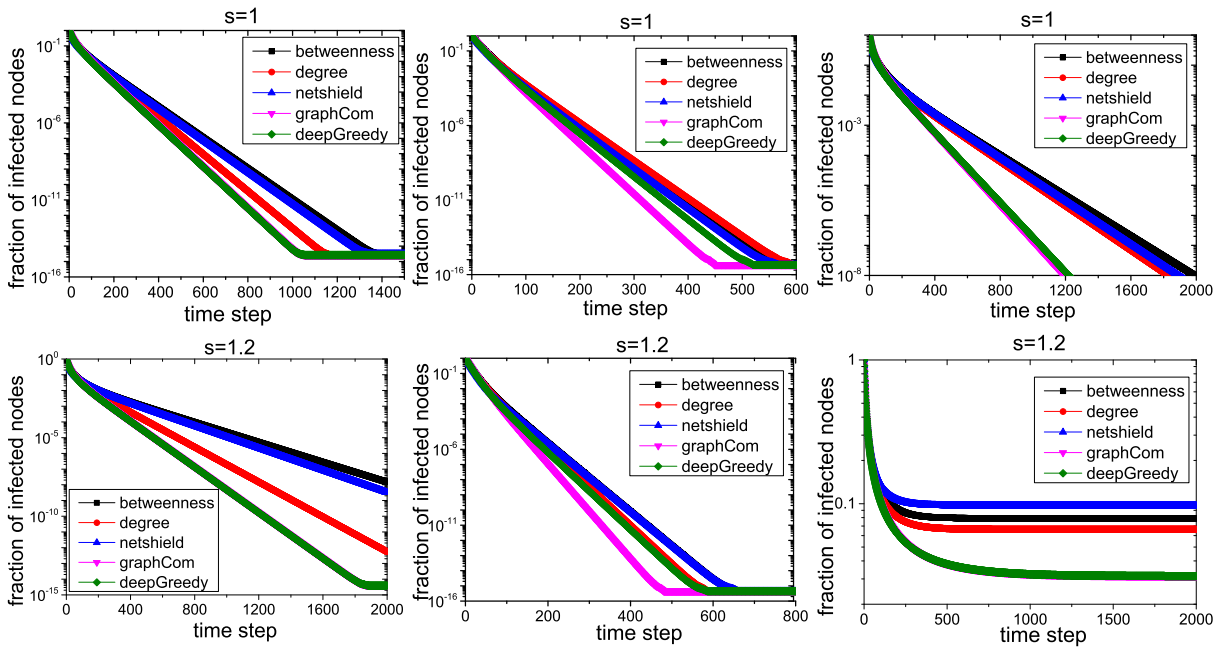


**Figure A5** The fraction of infected nodes vs. time step, (a)-(c) for s=1, (d)-(f) for s=1.2.

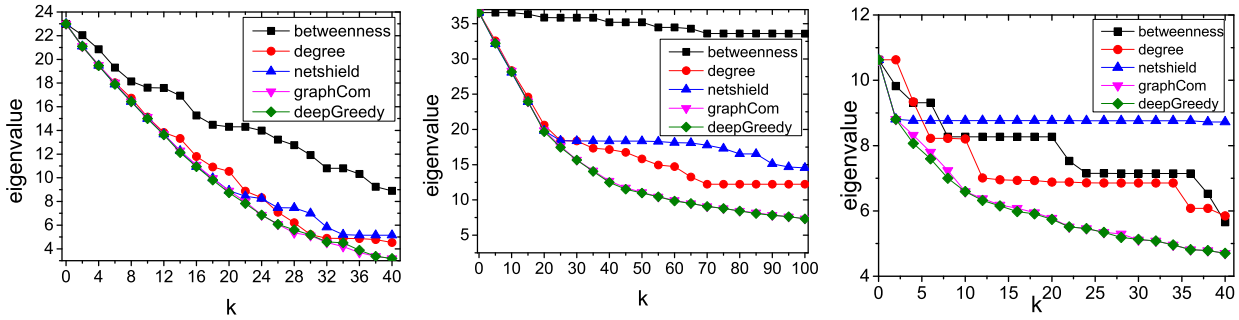


**Figure A6** Largest eigenvalue vs. the number of immune nodes $k$ in large network.

## References

1 Parchas P, Gullo F, Papadias D, et al. The pursuit of a good possible world: extracting representative instances of uncertain graphs[C]. In: Proceedings of ACM SIGMOD international conference on management of data, 2014: 967-978.

2 Tong H, Prakash B A, Tsourakakis C, et al. On the vulnerability of large graphs[C]. In: Proceedings of IEEE International Conference on Data Mining, 2010: 1091-1096.

3 Chen C, Tong H, Prakash B A, et al. Node immunization on large graphs: Theory and algorithms[J]. IEEE Trans Knowl Data Eng, 2016, 28(1): 113-126.

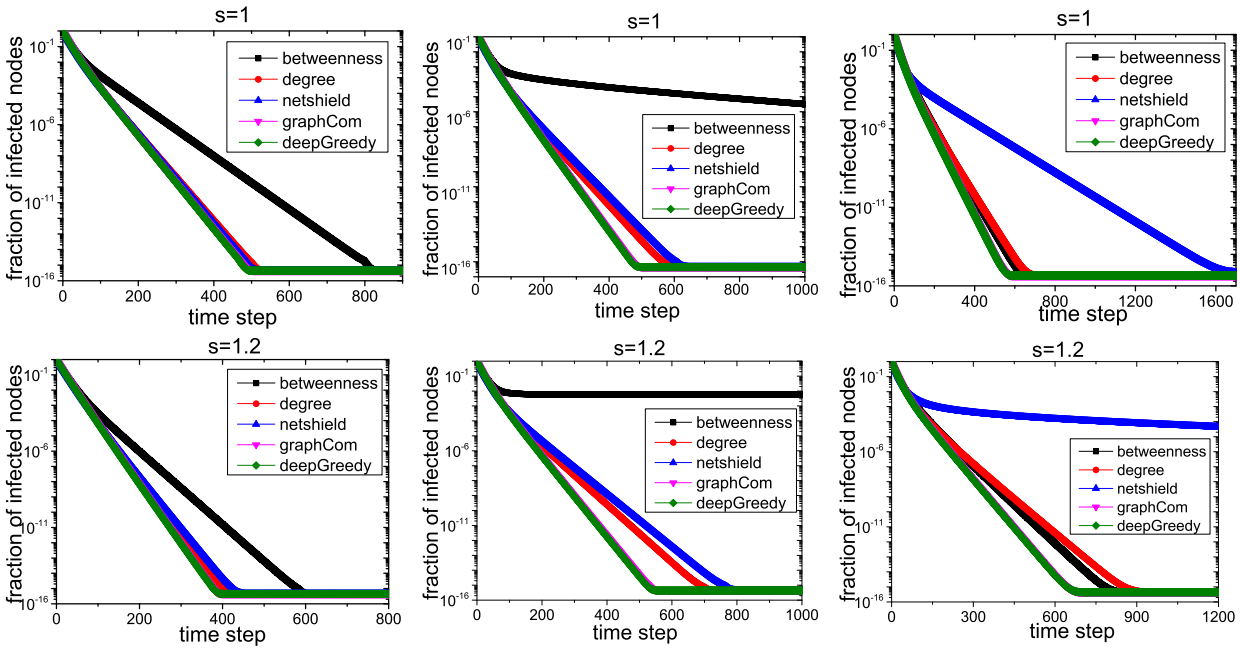


**Figure A7** The fraction on infected nodes in large networks.