Learning Vaccine Allocation from Simulations

Gerrit Großmann(B), Michael Backenköhler, Jonas Klesen, and Verena Wolf
Saarland Informatics Campus, Saarland University, 66123 Saarbrücken, Germany
{gerrit.grossmann,mosi}@unisaarland.de

Abstract. We address the problem of reducing the spread of an epidemic over a contact network by vaccinating a limited number of nodes that represent individuals or agents.

We propose a Simulation-based vaccine allocation method (Simba), a combination of (i) numerous repetitions of an efficient Monte-Carlo simulation, (ii) a PageRank-type influence analysis on an empirical transmission graph which is learned from the simulations, and (iii) discrete stochastic optimization.

Our method scales very well with the size of the network and is suitable for networks with millions of nodes. Moreover, in contrast to most approaches that are model-agnostic approaches and solely perform graph-analysis on the contact graph, the stochastic simulations explicitly take the exact diffusion dynamics of the epidemic into account. Thereby, we make our vaccination strategy sensitive to the specific clinical and transmission parameters of the epidemic.

Keywords: SIR Model · Vaccination allocation · Networked epidemic spreading · Control of epidemics · Network robustness and resilience

1 Introduction

Networks provide a universal language to represent interacting systems with emerging dynamical patterns. Computationally, every propagation process over a network can be considered an epidemic. Examples include actual pathogens on human contact networks [1], fake-news in online social networks [8,10], cascading failures in an infrastructure network [11], congestions in a traffic network, malware in computer networks [3], neural activity in a brain network [4], etc.

The problem of vaccine allocation is linked to the control of such a propagation, where limited vaccine resources are available and we aim to reduce the spread as much as possible, that is, lower the number of nodes reached by the epidemic.

Vaccine allocation strategies can help in the design of complex systems to make them more resilient against (cascading) failures. This is particularly relevant regarding infrastructure networks where a “vaccination” might represent the installation of some kind of protective safeguard. Another example is the mitigation of fake-news in online social networks which can be achieved by removing
the accounts of particularly relevant and malicious influencers or by providing warnings and fact-checking. In the context of infectious diseases, vaccination strategies are an appropriate way of setting priorities in vaccine distribution.

As a model for epidemic spreading, we consider the widely used stochastic, continuous-time Susceptible-Infected-Recovered (SIR) model \[9\]. Specifically, nodes (eventually) become immune after an infection (or die) and do not transmit the pathogen further. However, our framework is easily adaptable to epidemic models with more disease stages, such as COVID-19 models \[5\]. We consider as input an (undirected, unweighted) contact network with \(n\) nodes and a budget \(k\) (number of vaccines). The goal is to identify those \(k\) (susceptible) nodes which, when vaccinated, reduce the spread of the epidemic the most. We measure this by using the expected number of susceptible nodes in the terminal state, where the epidemic is over and all nodes are either susceptible, recovered, or vaccinated. In most cases, the only numerically feasible way to approximate this number is to perform a large number of stochastic simulations.

Generally speaking, the vaccination allocation problem is computationally difficult. Intuitively, it is often a good decision to vaccinate those nodes with a large number of neighbors (or with a high centrality in the network) and those which are close to the initially infected nodes. If possible, it is even better to identify those nodes which lie between the initially infected nodes and many susceptible nodes. If we represent the spreading process by a transmission tree (cf. Fig. 1), in which the direct children of a node \(v\) correspond to those nodes that were infected by \(v\), the size of a \(v\)'s subtree gives the number of multi-hop infections that originated from \(v\). The premise of our work is that the number of multi-hop infections of a node is a good indicator of whether that node is a good vaccination candidate.

Here, we propose Simba, (Simulation-based vaccine allocation), which is a method that makes use of recent developments in fast simulation of epidemic processes using a rejection-based approach \[6\]. This allows performing a large number of simulation runs of networks with millions of nodes and edges in the order of minutes on a standard desktop PC. Based on many simulations, Simba constructs a transmission graph, a generalization of the transmission tree for several simulation runs. By analyzing this graph, we obtain an impact score for every node. Repeated evaluation of the current vaccination strategy and re-computation of the impact scores yields an iterative optimization procedure, whose objective is to maximize the expected number of nodes that remain healthy.

The key methodological novelty of our proposed vaccination strategy is the construction and analysis of an empirical transmission graph. It poses a methodological framework to analyze contagion impact on complex networks. Using the transmission graph, our vaccination strategy can take the dynamics of the epidemic into account. The transmission graph has potentially many more use cases in assessing network dynamics, such as influence maximization, controllability of networks, impact/centrality quantification, and flow prediction. We also provide a numerical evaluation and compare Simba to several baselines from the literature.
The manuscript is organized as follows: We first provide a literate overview (Sect. 2), then we formalize the problem statement (Sect. 3). In Sect. 4, we introduce and explain our vaccination allocation method. Experimental results are presented in Sect. 5 and a conclusion completes the manuscript in Sect. 6.

2 Related Work

Traditionally, most methods focused on finding nodes for vaccination using a static analysis of the contact network, for instance by looking at the betweenness centrality of nodes [17] or at their degree [15]. Likewise, NetShield tries to minimize the epidemic threshold of the contact graph (i.e., its general ability to support epidemics) [20]. A more advanced method is GraphShield that starts with degree centrality but then takes the flow of information in the contact graph into account [21]. Eventually, researchers focused more on the dynamical aspects for instance by utilizing linear programming [16] or reinforcement learning [22,23]. For an overview, we refer the reader to [12].

Conceptually most relevant for us is the work of Zhang et al. who propose DAVA [24] and Song et al. who propose NIIP [18]. Both methods are based on a dominator tree architecture which tries to capture the direction of the epidemic. DAVA merges all initially infected nodes and analyzes the paths from this node to all other nodes. Nodes that block a large number of paths are suitable vaccination candidates. NIIP focuses on a problem setting where not all vaccination units are distributed at once. Therefore, NIIP extracts a maximum DAG from the contact graph and uses Monte-Carlo simulation to find the best nodes to vaccinate and combines this with a greedy simulation-based approach, the simulation’s goal is to determine when to distribute a vaccine.

3 Problem Statement

We first formalize the generative epidemic spreading model and the vaccination allocation problem. We remark that our framework can easily deal with all other types of spreading models as well.

3.1 Continuous-Time Networked SIR Model

Network State. Let $G = (V, E)$ be an undirected, unweighted graph with node set $V = \{v_1, \ldots, v_n\}$, containing $n$ nodes, and an edge set $E$ and let any $L : V \rightarrow \{S, I, R\}$ be a node labeling that assigns a node state to each node (corresponding to susceptible, infected and recovered nodes). We assume that $G$ is connected (i.e., all nodes are reachable from all other nodes) and has no self-loops. Each labeling function corresponds to a joint state (i.e., a superposition of all node states), called network state. The network dynamics specify how the network state (i.e., the labeling) changes over time. We use $L_{\text{init}}$ to denote the labeling of the initial network state and we use $S_{\text{init}}$ ($I_{\text{init}}$, $R_{\text{init}}$) to denote those nodes that were susceptible (infected, recovered) in $L_{\text{init}}$. 
Network Dynamics. We also assume that an infection rate constant $\lambda \in \mathbb{R}_{>0}$ and a recovery rate constant $\mu \in \mathbb{R}_{>0}$ are given. W.l.o.g. we typically assume $\mu = 1$ and only vary $\lambda$ as the spreading dynamics is determined by the fraction $\frac{\lambda}{\mu}$. The network state evolves according to a race condition between nodes and edges. Generally, all infected nodes can recover at rate $\mu$ and each $S - I$ edge can transmit an infection at rate $\lambda$, causing the susceptible node to become infected. Consequently, a (computationally naive but statistically correct) simulation run is performed by starting with $L_{init}$ and then, in each simulation step, drawing a firing time for each $I$-node and for each $S - I$ edge which are exponentially distributed with rates $\mu$ and $\lambda$, respectively. The event with the shortest firing time “wins” and the corresponding node state (label) is changed accordingly. Repeatedly applying these rules will always lead to a terminal labeling or network state where no more actions are possible (all nodes are recovered or susceptible). Given a random simulation run, we use the term transmission tree (cf. Fig. 1) to describe a tree where patient zero is the root (if there are more than one infected nodes in the beginning, we merge them) and every node that became infected during the course of the epidemic is connected to the node which infected it. Thus, all nodes in the subtree of a node are called its children, i.e. they were directly or indirectly infected by that node.

3.2 Vaccination Allocation Problem

We are given a finite contact network $G = (V,E)$ with corresponding initial labeling $L_{init}$, a vaccination budget $k \in \mathbb{Z}_{>0}$, as well as the infection and recovery rate constants $\lambda$ and $\mu$. We want to find a set $X$ of nodes to be vaccinated, where

$$X \subseteq S_{init} \quad \text{and} \quad |X| = k. \quad (1)$$

Moreover, for a given $G$, $L_{init}$, $k$, $\lambda$, $\mu$, we use $F(X)$ to denote the objective function which we define as the expected number of susceptible nodes in the terminal labeling when initially all nodes in $X$ are vaccinated. We define the
Vaccine Allocation Problem as:

Find a set $X$ that maximizes $F(X)$ such that (1) holds.

In practice, we approximate $F(\cdot)$ statistically based on many Monte-Carlo simulation runs. We assume that at least $k$ nodes exist that can be vaccinated and there is at least one infected node in the initial labeling. We model the vaccination by setting $L_{\text{init}}(v) = R$ for all $v \in X$ at the beginning of the simulation. Note that (assuming the vaccination works perfectly) already recovered, deceased, and vaccinated nodes do not differ from the simulation’s point of view.

**Complexity.** The problem is computationally difficult because there are $\binom{n}{k}$ possibilities to distribute $k$ vaccines to $n$ nodes. The corresponding decision problem is $\mathcal{NP}$-hard. Specifically, for a given input $G$, $L_{\text{init}}$, $\lambda$, $\mu$, and threshold $\tau$, it is $\mathcal{NP}$-hard (in $n$) to decide if a solution $X$ exists s.t. $F(X) > \tau$. It can be shown that for this type of problem, $\mathcal{NP}$-hardness holds for any propagation model that can mimic an independent cascade (IC) model [24]. We can do this by making $\mu$ ($\lambda$) arbitrary small (large). Note that this only holds for the SIR model and not necessarily for the generalizations to arbitrary spreading models.

4 Our Method

We first explain the main components of Simba (Simulation-based vaccine allocation): the rejection-based simulation method and the construction of the transmission graph with the identification of high-impact nodes based on a ranking analysis.

4.1 Rejection-Based Simulation

For our method, we exploit previous work on fast simulations to efficiently perform a large number of simulations [2,6]. We propose an algorithm that is statistically equivalent to the generative process description in Sect. 3.1. We perform event-driven simulation using a priority queue. For the initialization, we create one recovery event and one infection event for each infected node and push them into the queue. The firing time is exponentially distributed with rate $\mu$ (recovery event) and rate $\lambda \cdot d_i$ (infection from node $v_i$, $d_i$ being the number of $v_i$’s neighbors). In each simulation step, we take the first event from the queue. If it is a recovery event, we simply set the corresponding node to state $R$. If it is an infection event, we first check if the corresponding node is still in state $I$, if not, we reject the event and proceed with the next step. If it is, we pick a random neighbor, which will be the target of the infection. We check if the random neighbor is susceptible. If it is, we set the neighbor to $I$ and create two events (recovery and infection) for the newly infected neighbor. We also create a new infection event for the source node. Then, we proceed with the next step. The simulation ends when there are no more nodes in state $I$. 
Fig. 2. Schematic illustration of the transmission graph construction based on the same setting as Fig. 1. We consider 10 simulation runs. Center: Contact graph with \( I_i \) and \( I_{(i,j)} \) as node and edge labels, respectively. Right: Adding the dummy node and normalizing outgoing weights yields a discrete-time Markov chain (DTMC). Nodes in \( S_{\text{init}} \) are annotated with their impact score based on the equilibrium of the DTMC.

We store the number of susceptible nodes when the simulation ends. Moreover, each time a node gets infected, we store from which (infected) neighbor the infection originated (or all nodes it could have originated from, cf. Sect. 4.2).

### 4.2 Impact Score Estimation

To estimate the nodes’ impacts, we build an empirical transmission graph (cf. Fig. 2), an extension of the transmission tree from Fig. 1 to multiple simulation runs. The transmission graph is directed and one can perform a random walk on the graph which (on average) visits nodes with higher impact more often. In the end, we determine the impact of each node in \( S_{\text{init}} \) by ranking the nodes similar to the idea of the well-known PageRank [13] (i.e., the equilibrium of the corresponding Markov chain).

Given a set of simulated trajectories, let \( I_i \) denote the number of trajectories in which node \( v_i \) became infected. Furthermore, let \( I_{(i,j)} \) denote the number of trajectories in which \( v_i \) directly infected \( v_j \). Note that \( I_i = \sum_j I_{(j,i)} \).

**Transmission Graph.** We construct a transmission graph \( G_T = (V_T, W_T) \) (with \( W_T \) being a weight matrix) as follows: We start with a dummy node \( v_D \) as a sink, that is \( V_T = V \cup \{v_D\} \) (we can remove unreachable nodes later), and add an edge from each initially infected node with weight one, i.e. for all \( i \):

\[
W_T(i, D) = \begin{cases} 
1 & \text{if } v_i \in I_{\text{init}}, \\
0 & \text{otherwise}.
\end{cases}
\]
Then we add an edge from \( v_D \) to each \( v_i \in S_{\text{init}} \) with a weight proportional to the estimated probability of that node becoming infected, i.e. for all \( i \):

\[
W_T(D, i) = \begin{cases} 
\frac{I_i}{\sum_{v_j \in S_{\text{init}}} I_j} & \text{if } v_i \in S_{\text{init}}, \\
0 & \text{otherwise.}
\end{cases}
\]

Moreover, for all nodes \( v_i \neq v_D \) (we consider \( 0/0 \) as 0):

\[
W_T(i, j) = \frac{I(j,i)}{I_i}.
\]

Note that, by construction, the outgoing weights in \( G_T \) are normalized and therefore represent a discrete-time Markov chain. A random walk in the chain will preferably visit nodes of high impact on the epidemic as the transition probabilities are proportional to the estimated infection probabilities. We compute the equilibrium distribution of the corresponding Markov chain using the power iteration method [19]. We call the equilibrium probability of a node normalized over \( S_{\text{init}} \) its impact score. We only consider the impact score for nodes in \( S_{\text{init}} \) because only those are eligible for vaccination. Note that a transmission graph for a single simulation run is equivalent to the transmission tree where all edges have weight one.

We can make the transmission graph even more accurate. During the simulation, instead of only storing the node that actually transmitted the infection, we store all neighbors that could have potentially been the source of the infection. In our model, each infected neighbor was equally likely to have transmitted the pathogen. It is straightforward to adapt the construction of the transmission graph accordingly even in non-Markovian settings.

### 4.3 Introducing Simba

We combine the efficient simulations with the transmission graph analysis with an iterative optimization scheme to arrive at Simba.

**Greedy Initialization.** We use \( C_i \) to denote the set of vaccinated nodes in iteration \( i \). We start with an empty set, \( C_0 \), of nodes to be vaccinated. Until \( |C_i| = k \), we compute the impact score for all nodes \( v_i \in S_{\text{init}} \) (assuming nodes in \( C_i \) are vaccinated) and add the node with the highest impact to \( C_i \), leading to \( C_{i+1} \).

**Optimization.** In each optimization step \( i \), we randomly remove one node (with equal probability) from \( C_i \) (leading to set \( B_i \)) and compute the impact score of all nodes \( v_i \in S_{\text{init}} \setminus C_i \) (assuming nodes in \( B_i \) are vaccinated). Then we add one of the nodes with the highest impact to \( B_i \) (nodes with higher impact are more likely to be chosen), leading to \( C_{i+1} \). We estimate \( F(C_i) \) in each iteration step and repeat until some stopping criterion is reached, then we return the set that yielded the highest estimated score.
Assume we want to estimate the impact of the two successor nodes of patient zero based on two simulation runs. Using, for example, the size of their corresponding subtree in the transmission tree leads to misleading results in this case. Specifically, both nodes would be assigned drastically different values. Combining the two runs in a transmission graph yields a more realistic impact score than considering both runs separately. Note that edges point to the origin of the infection and the transmission graph is shown without its dummy node.

### 4.4 Discussion

Here, we want to address three non-obvious questions: (i) *why build a transmission graph?*, (ii) *what does the graph say about the objective function?*, and (iii) *why is it necessary to consider the dynamics at all?*

#### Building a Transmission Graph.

Using the transmission graph has multiple advantages. Most importantly, transmission trees only mimic a subset of possible infection flows. In contrast, transmission graphs make it possible to aggregate information over many runs in a principled manner (cf. example in Fig. 3). This way they capture the interplay between connectivity and infection flow more precisely. Moreover, computing the equilibrium of the Markov chain is computationally fast and theoretically well principled. It is also possible to efficiently build the transmission graph on-the-fly during the simulations.

#### Impact Score and Objective.

Note that we handle two different problems. The impact score quantifies the question “*How many nodes became infected as a direct (‘multi-hop’) consequence from each node?*” However, the objective \( F(\cdot) \) is concerned with “*How many nodes will (on average) not become infected if a specific (set of) node(s) is vaccinated?*” The latter question is notoriously more difficult to answer. The reason why they differ is that if we vaccinate a node, all of its children in the transmission tree can still become infected via alternative paths. In this sense, the impact score gives an over-approximation on the effect of vaccinating a node regarding the objective. Colloquially, if we vaccinate a node with \( m \) children (on average), then the best we can hope for is that these \( m \) nodes do not become infected. Hence, our optimization procedure picks nodes depending on their theoretical (and over-approximated) capability or potential to reduce the epidemic spreading.
Fig. 4. Assume $k = 1$. The best node to vaccinate depends on the dynamics. If $\lambda$ is small, the infection will die out on its own in the line graph and it makes more sense to protect the FCC even though it contains fewer nodes. The opacity illustrates a node’s probability to become infected. The nodes are numbered in decreasing order of their impact scores.

**Importance of Dynamics.** The goal is to vaccinate nodes such that the network becomes less “supportive” of epidemics spreading in it. But then why should the specific dynamics matter? In other words, how can vaccinate a specific node be the right decision for some infection rate constants and the wrong decision for other ones? It is easy to see this in the example in Fig. 4 where we have a single patient zero and a budget of $k = 1$. We can either vaccinate the node to the “right” to protect the fully connected component (FCC) with six nodes or we can vaccinate the node to the “left” to protect the line-graph with nine nodes. If the epidemic is “weak”, it will die out anyway over the line graph, so it makes sense to protect the FCC. In contrast, protecting the line-graph “saves” more nodes if the epidemic is strong enough to conquer the whole graph.

**4.5 Generalizations**

Our framework can easily be extended to various epidemic-type models. The only necessity is that (i) the model can be simulated (efficiently), (ii) there is a clear objective (e.g., maximize susceptible nodes in terminal states), and (iii) the process represents some contagion phenomena (such that the transmission graph can capture a direction of the information flow). Potential generalizations include models with more disease stages (like SEIR), non-Markovian dynamics (e.g., where the infectiousness of nodes changes over time), weighted and directed networks, as well as temporal or adaptive networks and time-discrete models. Simba can also be adapted to different objectives. For instance, in the SIS model (where infected nodes become susceptible again) the goal is typically to minimize the number of infected nodes in the equilibrium. In that case, our method would identify the nodes that are generally most impactful for the epidemic spreading and not only with regards to a specific initial set of infected nodes. Likewise, we could optimize the timepoints of vaccine distribution [18]. Simba can also be used when the transmission parameters are unknown by using an infection rate constant slightly above the epidemic threshold [14].
Fig. 5. Left: Optimization of the terminal fraction of expected susceptible nodes $F(X)$ on three sample networks. Right: Runtime of 1000 simulations and solution of the corresponding transmission graph based on random $d$-regular graphs.

5 Experimental Results

We provide an implementation of Simba in Rust and Python (for visualization/IO), code will be made available\footnote{github.com/gerritgr/Simba}. We used synthetic networks following three random graph models (Erdős-Rényi, Geometric, and Barabási-Albert (BA)) with $10^2$, $10^3$, and $10^4$ nodes, respectively. The corresponding budgets are $k = 2$, $k = 3$, and $k = 10$. We consider the following baselines: random (expected $F(X)$ when random nodes are vaccinated), DAVA, and DAVA-fast\cite{24}, PageRank, and Pers. PageRank (personalized PageRank)\cite{7,24}, and Degree (pick nodes with highest degree). We use $10^3$ simulations runs for each construction of the transmission tree. We also analyze the runtime of a complete construction and solution of a transmission graph based on $d$-regular random graphs (i.e., all nodes have exactly $d$ neighbors) with varying degree $d$ and $n$. Practically, the runtime is almost linear in $n$. Theoretically, the number of simulation steps in each run increases linearly. The costs of each simulation step increase sub-linearly. The costs of solving the DTMC also increase linearly. We see that, even though the number of iteration steps is quite small, Simba is superior to or (almost) on par with the baselines in the experiments. Simba struggles the most with BA graph which is a special case but important as it highlights potential problems. It seems that the general strategy of Simba to separate the initially infected from the susceptible nodes does not work better than identifying the nodes which are generally important for the graph’s resilience against epidemics. This is due to the fact that BA graphs typically possess a small subset of nodes that are extremely effective candidates for vaccination regardless of the infection source. Note that DAVA also struggles in this case while Degree and both PageRank methods shine.

6 Conclusions and Future Work

We presented a novel technique to find the most suitable vaccination candidates in a network. Unlike other methods, our approach is based on statistically correct
simulations which are analyzed using the transmission graph. The transmission graph represents the flow of a pathogen in the network as a directed weighted graph. The method is suitable for all epidemic models that can be simulated efficiently.

In the future, we aim to perform different types of information flow analysis on the transmission graph, not only random walks. It remains to be determined which kind of flow analysis is most useful for which type of objective (e.g., vaccination, control, influence maximization). Moreover, we want to extend numerical evaluations to more complex spreading models (e.g., non-Markovian, multi-state ones) and network types (e.g., adaptive networks).

**Acknowledgments.** This work was partially funded by the DFG project MULTI-MODE.

**References**

1. Ball, F., Sirl, D., Trapman, P.: Analysis of a stochastic sir epidemic on a random network incorporating household structure. Math. Biosci. 224(2), 53–73 (2010)
2. Cota, W., Ferreira, S.C.: Optimized Gillespie algorithms for the simulation of Markovian epidemic processes on large and heterogeneous networks. Comput. Phys. Commun. 219, 303–312 (2017)
3. Gan, C., Yang, X., Liu, W., Zhu, Q., Zhang, X.: Propagation of computer virus under human intervention: a dynamical model. Discrete Dynam. Natu. Soc. 2012 (2012)
4. Goltsev, A., De Abreu, F., Dorogovtsev, S., Mendes, J.: Stochastic cellular automata model of neural networks. Phys. Rev. E 81(6), 061921 (2010)
5. Grossmann, G., Backenköhler, M., Wolf, V.: Importance of interaction structure and stochasticity for epidemic spreading: a covid-19 case study. ResearchGate (2020). https://www.researchgate.net/publication/341119247_Importance_of_Interaction_Structure_and_Stochasticity_for_Epidemic_Spreading_A_COVID-19_Case_Study
6. Großmann, G., Wolf, V.: Rejection-based simulation of stochastic spreading processes on complex networks. In: International Workshop on Hybrid Systems Biology, pp. 63–79. Springer (2019)
7. Jeh, G., Widom, J.: Scaling personalized web search. In: Proceedings of the 12th International Conference on World Wide Web, pp. 271–279 (2003)
8. Khurana, P., Kumar, D.: Sir model for fake news spreading through whatsapp. In: Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT), pp. 26–27 (2018)
9. Kiss, I.Z., Miller, J.C., Simon, P.L., et al.: Mathematics of Epidemics on Networks. vol. 598. Springer, Cham (2017)
10. Kitsak, M., Gallos, L.K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H.E., Makse, H.A.: Identification of influential spreaders in complex networks. Nat. Phys. 6(11), 888 (2010)
11. May, R.M., Arinaminpathy, N.: Systemic risk: the dynamics of model banking systems. J. R. Soc. Interface 7(46), 823–838 (2009)
12. Nowzari, C., Preciado, V.M., Pappas, G.J.: Analysis and control of epidemics: a survey of spreading processes on complex networks. IEEE Control Syst. Mag. 36(1), 26–46 (2016)
13. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report 1999-66, Stanford InfoLab, November 1999. http://ilpubs.stanford.edu:8090/422/, previous number = SIDL-WP-1999-0120

14. Prakash, B.A., Chakrabarti, D., Valler, N.C., Faloutsos, M., Faloutsos, C.: Threshold conditions for arbitrary cascade models on arbitrary networks. Knowl. Inf. Syst. 33(3), 549–575 (2012)

15. Prakash, B.A., Tong, H., Valler, N., Faloutsos, M., Faloutsos, C.: Virus propagation on time-varying networks: theory and immunization algorithms. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 99–114. Springer (2010)

16. Sambaturu, P., Vullikanti, A.: Designing robust interventions to control epidemic outbreaks. In: International Conference on Complex Networks and Their Applications, pp. 469–480. Springer (2019)

17. Schneider, C.M., Mihaljev, T., Havlin, S., Herrmann, H.J.: Suppressing epidemics with a limited amount of immunization units. Phys. Rev. E 84(6), 061911 (2011)

18. Song, C., Hsu, W., Lee, M.L.: Node immunization over infectious period. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 831–840 (2015)

19. Stewart, G., Miller, J.: Methods of simultaneous iteration for calculating eigenvectors of matrices. Top. Numer. Anal. II, 169–185 (1975)

20. Tong, H., Prakash, B.A., Tsourakakis, C., Eliassi-Rad, T., Faloutsos, C., Chau, D.H.: On the vulnerability of large graphs. In: 2010 IEEE International Conference on Data Mining, pp. 1091–1096. IEEE (2010)

21. Wijayanto, A.W., Murata, T.: Flow-aware vertex protection strategy on large social networks. In: 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 58–63. IEEE (2017)

22. Wijayanto, A.W., Murata, T.: Learning adaptive graph protection strategy on dynamic networks via reinforcement learning. In: 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 534–539. IEEE (2018)

23. Wijayanto, A.W., Murata, T.: Effective and scalable methods for graph protection strategies against epidemics on dynamic networks. Appl. Netw. Sci. 4(1), 18 (2019)

24. Zhang, Y., Prakash, B.A.: Data-aware vaccine allocation over large networks. ACM Trans. Knowl. Discov. Data (TKDD) 10(2), 1–32 (2015)