# Identifying influential spreaders on a weighted network using HookeRank method

Sanjay Kumar[1,2], Nipun Aggarwal[2], and B.S. Panda[1]

[1] Computer Science and Application Group, Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India.
[2] Department of Computer Science and Engineering, Delhi Technological University, Main Bawana Road, New Delhi 110042, India

**Abstract.** Influence maximization is a significant research problem that requires the selection of influential users who are capable of spreading information in the network such that it can reach to a large number of people. Many real-world networks like Road Network, Email Networks are weighted networks. Influence maximization on weighted networks is more challenging than an unweighted network. Many methods, such as weighted-degree rank, weighted-voteRank, weighted-eigenvalue rank, and weighted-betweenness rank methods, have been used to rank the nodes in weighted networks with certain limitations. In this manuscript, we propose a Hooke's law-based approach named HookeRank method to identify spreaders in a weighted network. We model edge weights as spring constants. The edges present in the network are modeled as springs, which are connected in series and parallel. They elongate by a distance under the effect of a given constant force following Hooke's law of elasticity, and this is the equivalent propagation distance between nodes in the network. The proposed the model finds relevant influential nodes, that can propagate the information to other nodes. A higher HookeRank score implies the greater influential capability of the node in the network. We compared our proposed algorithm with state-of-the-art models and found that it performs reasonably well on real-life data-sets using epidemic spreading Susceptible-Infected-Recovered model.

**Keywords:** Influence Maximization· Hooke's Law· Information Diffusion· Social Networks· Weighted Networks

## 1 Introduction

Many real-world networks, like online social networks, transportation networks, email networks, collaboration networks, and many others are complex weighted networks [1]. These networks can be modeled as graphs, $G = (V, E, W)$ where $V$ represents a set of nodes, $E$ denotes edges between nodes, and $W$ represents the edge weight. The weight of a connection between two nodes usually depends on the exchange of services, intensity, or duration [2]. If two nodes are frequently interacting or if they have high interactions, then information or diseases are more likely to be transferred between them [3]. Complex networks have a large

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

number of nodes, and interaction between nodes is usually complicated. The evolution of complex networks has led to the establishment of many useful applications like influence maximization, viral-marketing, and information propagation [4]. Influence maximization [5] is a technique to select some constant number of nodes as seed nodes which are capable of spreading information by "Word-of-Mouth" analogy, after knowing the information source. Therefore, influence maximization finds great value in the business. The influence maximization has many applications, for example, controlling the proliferation of messages and rumors, positioning influential researchers, and discovering social leaders. The process of influence maximization consists of two activities, first, identifying the seed spreaders and the second the information diffusion phase. In the study of disease transmission, numerical models are helpful in understanding the spread and control of epidemics. The circulation of the information in the complex network is very similar to the epidemic spreading, and many popular methods for modeling information diffusion are based on the epidemics spreading [6,7]. In epidemiology, mathematical models play a role as a tool in analyzing the spread and control of infectious diseases [8]. Many researchers have successfully applied epidemic spreading models to information propagation in complex networks to estimate the final spread of the information originating from the source nodes.

Most of the influence maximization models on weighted-networks are merely extensions of the algorithms counterparts on unweighted networks by introducing edge weight into the models. Numerous other models have considered standard graph-theoretical features based approach for identifying the important spreaders. In this paper, we propose a Hooke's law of elasticity based approach named HookeRank method to identify spreaders in a weighted network. Our algorithm considers the influence of nodes in a setting where edges are modelled as springs and edge-weights are modelled as elasticity coefficients. We model edge weights as spring constants. The edges present in the network are modeled as springs, which are connected in series and parallel. They elongate by a distance under the effect of an assumed constant force following Hooke's law of elasticity, and this is the equivalent propagation distance between nodes in the network. The contributions of our work are as follows:

1. We propose a novel method based on Hooke's Law of Elasticity in complex weighted networks to find the influential spreaders.
2. We model the equivalent weight between indirectly connected nodes in a weighted network.
3. The proposed algorithm is an improved method of selection of influential nodes on real-world data-sets.

The rest of the paper is organized as follows: Section 2 consists of the related work in this field. Section 3 presents the data-sets and the models for information diffusion used in this paper. In Section 4, we describe the methodology of our novel method and simulation on a toy network. Section 5 discusses the simulation of our algorithms on various real-world networks. Finally, Section 6 concludes our paper.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

## 2 Traditional Centralities

The initial models in the field of influence maximization have majorly been innovations in the field of unweighted networks where all edges are equally important. In real-world networks, these edges are associated with weights that need to be considered while analyzing the strength of these connections, during a cycle of information diffusion. When we consider these aspects of topology, we can gather insights into what is most beneficial for the maximization of information diffusion. The early advances in weighted networks were through centralities like DegreeRank used for unweighted networks, by additional weighing of these edges to achieve a weighted DegreeRank [9]. In a similar pattern, multiple centralities were eventually derived from unweighted networks, evolving into a method for weighted graphs through mathematical adjustments leading to weighted algorithms. Betweenness centrality considers the shortest path of a node in an unweighted graph, and it was extended for weighted version giving the weighted-betweenness centrality [10,11]. Based on the notion of the voting scheme, researchers have proposed influence maximization algorithm in unweighted as well as weighted networks where the nodes getting the highest votes in each round gets selected as spreader nodes [12,13,14]. The $h$-index is a measure of the impact of researchers based on the number of citations received, and by augmenting edge weight, Yu et al. proposed a weighted $h$-index centrality [15]. Weighted-eigenvector centrality applicable in a weighted network is based on the fact that a node is important if its neighbors are also famous and finds the centrality for a node as a function of the centrality of its neighbors [16]. Eades [17] suggested to model the edges of the network as springs to draw graphs by minimizing potential energy. This method was later refined by Fruchterman et al. [18], where they model nodes as electrical charges and edges as connecting springs. The electrical charges make these nodes repel each other. One of the most popular algorithms for drawing graphs is Kamada and Kawai's method, which models the edges of the graph as springs acting following Hooke's Law [19,20]. The method optimizes the length of the spring between any two nodes by minimizing a global cost function. We argue the applicability of the spring-based model to measure the centrality of the nodes and to find influential spreaders.

## 3 Datasets and Performance Metrics

### 3.1 Information Diffusion Model

SIR Model: In this paper, we utilize the susceptible-infected-recovered (SIR) model as the data diffusion model [21]. This model divides nodes into three categories Susceptible ($S$), infected ($I$), and recovered ($R$). Susceptible nodes are supposed to receive data from their infected neighbor nodes. The information starts from a subset of the network nodes with the spreading parameter ($\beta$), and recovery rate ($\gamma$). In the SIR model, initially, all nodes, except seed nodes, are in a susceptible state. After each progression, the infected nodes affect their susceptible neighbors with a likelihood of $\beta$. Infected nodes at the next timestamp

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

enter the recovered stage with a likelihood of $\gamma$. When arriving at the recovered stage, they are no longer prone and can't be infected again.

### 3.2 Performance Metrics

The final infected scale ($f(t_c)$):- It is a measure of the final spread of the information originating from the chosen seed nodes at the end of SIR simulations. The final infected scale is the final number of recovered users that passed through the chronological advancements from susceptible, infected, and finally, to recover during the information diffusion process. There are two ways to measure this criterion, first $f(t_c)$ is plotted against time $t$, which shows us the propagation of the information on the network as time proceeds. Secondly, $f(t_c)$ is plotted against the different fraction of spreaders, which shows us the propagation of the information on the network as the number of spreaders taken by the algorithm initially is changed.

### 3.3 Datasets used

**Table 1.** Real world data-sets for simulation

| Dataset Name | Description | Nodes | Edges |
|---|---|---|---|
| PowerGrid [22] | An undirected weighted network containing information about the power grid of the Western States of the United States of America. | 4941 | 6954 |
| Facebook-like weighted network [23] | This undirected weighted dataset originates from an online community for students at the University of California, Irvine | 1899 | 20297 |
| US Airports [24] | An undirected weighted network of the 500 busiest commercial airports in the United States. | 500 | 28237 |

## 4 Methodologies

In this section, we present the mapping of the edges present in the network as springs, which are connected in series and parallel, and describe the proposed HookRank method. The discrete part of individual connections of springs is discussed in detail. In a weighted network, weights generally mean that the higher the weight, the stronger is the connection. The same is true for our network as well since we know from Hooke's law that more is the spring constant, less is the displacement from the spring. Now this new unit of distance between any two Nodes. This distance is the actual distance between these nodes when the

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

information diffusion is to be considered. By normalizing these distances, we can model not only the approximate form but also the equivalent of all the different paths that exist between any two pairs of nodes. When we model this, individual graphs for each node are created, and these nodes can now be evaluated based on the amount of information they can propagate. We will consider that a constant force $F_0 = 1$ continuously acts on the node that is chosen as a seed node. Now the seed node is connected to every other node with a spring constant of $k = w_{ij}$ where k is the spring constant of a spring that connects node $i$ and node $j$ with a weight of $w_{ij}$. Now using a breadth-first traversal (BFS), calculate the new spring between the edges, if there exist, multiple edges between the springs, on different levels, they must be added since they are definitely parallel. When traversing from the node of one level to another, use the series combination to generate equivalent springs and to calculate most probable distances and finally propagating information through each of these nodes to find the maximum amount of spreading that takes place.

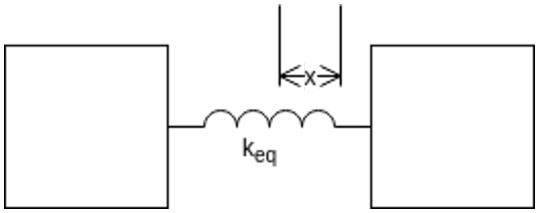


**Fig. 1.** Calculating the heuristic distance between two nodes based on weights of the edges, being modeled using springs and evaluating to a single spring, following Hooke's Law of Elasticity

**Parallel:** When Springs are placed in parallel, they end up as a joint spring with the total elasticity of a new spring of a spring constant that can be modeled using the fact that the spring is definitely much stiffer.

**Series:** It is possible to add the contributions of the springs in series. The Springs in series make a more flexible spring that tends to elongate more than the springs that are previously involved in the complete connection.

### 4.1 Distance Calculation

When two springs of different spring constants, $k_1$ and $k_2$ respectively are placed in series with each other, we get:

$$1/k_{eq} = 1/k_1 + 1/k_2 \tag{1}$$

When two springs of different spring constants, $k_1$ and $k_2$ respectively are placed

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

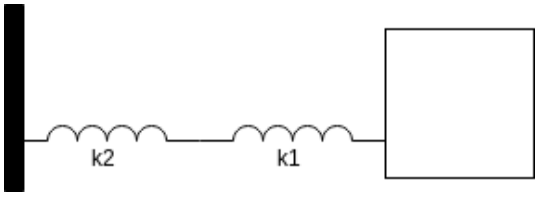


**Fig. 2.** Calculating the heuristic distance between two nodes based on weights of the edges, being modeled using springs and evaluating to a single spring, following Hooke's Law of Elasticity

in parallel with each other, we get:

$$k_{eq} = k_1 + k_2 \tag{2}$$

This means in an actual network is that springs in series are stiffer if the

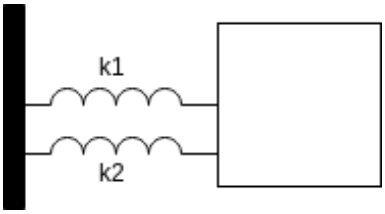


**Fig. 3.** Calculating the heuristic distance between two nodes based on weights of the edges, being modeled using springs and evaluating to a single spring, following Hooke's Law of Elasticity

strength of ties in the individual connections is strong. It also implies that more connections from one node to another, add up a single connection, as seen in case of parallel connections. Now the equivalent distance between any of these nodes, under a constant force is given as:

$$x = f/k \tag{3}$$

Assuming $f = 1$, without loss of generality, we can easily see that the full measure of distance in this network is relative. A breadth-first search similar method is used for the nodes to find the equivalent value of $k$ between all indirect neighbors.

### 4.2 Proposed algorithm

Based on the notion of modeling edges as springs, we compute the centrality of the nodes in the network. Edge weights in the network are the spring-constants

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

of these modeled springs. The proposed HookeRank method uses the following steps:

1. For each node, we perform a breadth-first search to all other nodes and calculate the distance of each node in series from the nearest neighbor.
2. In the BFS traversal, when a new node is encountered, we add its distance from its parent using a series combination as in equation 2 and add this neighbor to the queue.
3. All the nodes that occur again in the BFS traversal are assumed to be in a parallel connections and add up to the spring constant according to equation 1.
4. When for a given node, the queue is completely processed, we get its equivalent tree.
5. Calculate the HookeRank value for the node by finding the weighted average degree of the given node in the equivalent tree.
6. This procedure gives us the HookeRank value for each node, and the result is stored in a dictionary containing the node and its HookeRank value.
7. As, the objective of influence maximization is to select top $c$ nodes, where $c$ is a constant. Here, the top $c$ nodes are the nodes having the maximum HookeRank score values in the ranking, and such nodes can be chosen as the influential spreaders.

### 4.3 Time complexity of the proposed algorithm

The time complexity of the HookeRank method consists of initialization of spring constants and selection of the node with the highest number of the closest neighbor score and finding the level order traversal with respect to all the nodes (Step 1 to Step 4). Overall this makes the complete algorithm bounded by $O(V(E+V))$. This time complexity, however, reduces because we know that if an equivalent spring from $A$ to $B$ has the constant $k$, then spring from $B$ to $A$ has the same spring constant (Step 6).

## 5 Results and Analysis

We perform the experiment of the proposed HookeRank method along with the contemporary centrality measures like weighted- degree, weighted betweenness centrality, weighted eigenvector centrality, and weighted voteRank. The investigation has been performed on a toy network and three real-world networks of different nature, application, and size that are listed in Tab 3.3. We use the SIR model to compute the final infected scale, $f(t_c)$, as a function of spreaders fraction and final infected scale in terms of increasing timestamps. The results were averaged over SIR 100 simulations. For simplicity and to maintain consistency in the analysis for all data-sets, we chose infection rate ($\beta$ ) as 0.01, meaning that when a node is infected, then it can infect 1% of its neighbors randomly.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

### 5.1 Simulation of the proposed algorithm on a toy network

Here, we simulate the working of the proposed HookeRank method using a toy network, as depicted in Fig. 5.1. The network is a weighted graph with edge weights representing the stiffness constant of the spring.

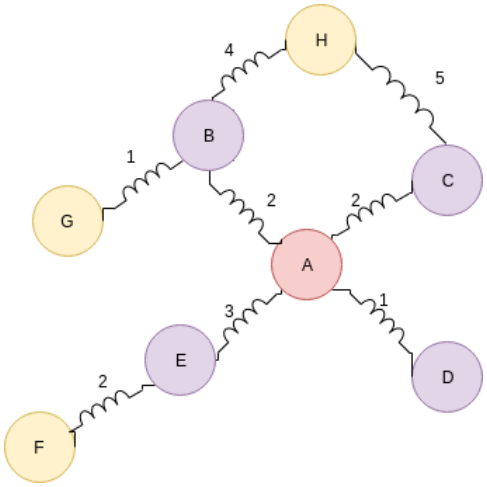


**Fig. 4.** A sample Weighted Network where the edges are modelled as springs and the spring constant is analogous to the weight of the edge

Let us consider the steps of the algorithm for a sample node $A$ to understand the working of this algorithm. We take a FIFO queue and put A inside the queue. Now, the distance to $A$ is 0. Now let's compute the spring constant for the first neighbors $B$, $C$, $D$, $E$ using the direct connection of the spring. These are then popped out, and their neighbors are pushed into the queue. The equivalent spring constant is found through a series connection through the parent. In the case of multiple parents at the same level, the constant is found using a parallel combination, as in the case of node $H$. For the node $A$, the simulation and

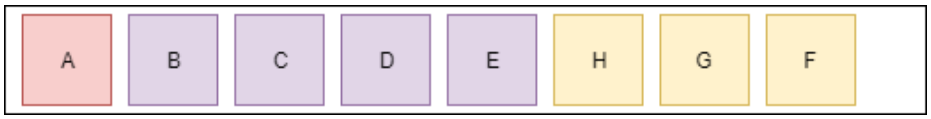


**Fig. 5.** The run of a breadth-first search with respect to A as the starting node

calculation of the equivalent distances of all the other nodes by using a breadth-first search are performed. Its immediate neighbors are processed first, and so on, the different layers are highlighted in a level order fashion, as shown in Fig.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

5. We can now compute the value of all the neighbors of $A$, as performed in the

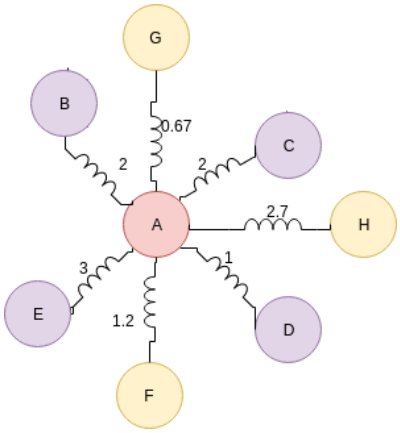


**Fig. 6.** The final network with respect to node $A$. Notice that certain indirect neighbors became direct neighbors under the action of both series and parallel connections.

above section. The computation will result in a graph similar to Fig. 5.1. Notice the connections and the spring constants. Thus, the average spring constant for $A$ becomes, *12.57 / 8 = 1.52*, and thus, an elongation averaging 1.57 are taken, giving us a HookeRank value of 1.57 for the node $A$.

**Table 2.** Results of HookeRank Score of various nodes in the toy Network

| Node | A | B | C | D | E | F | G | H | HookeRank-value |
|---|---|---|---|---|---|---|---|---|---|
| A | 0.0 | 2.0 | 2.0 | 1.0 | 3.0 | 1.2 | 0.67 | 2.7 | 1.57 |
| B | 2.0 | 0.0 | 3.2 | 0.67 | 1.2 | 0.75 | 1.0 | 4.0 | 1.60 |
| C | 2.0 | 3.2 | 0.0 | 0.67 | 1.2 | 0.75 | 0.76 | 5.0 | 1.69 |
| D | 1.0 | 0.67 | 0.67 | 0 | 0.75 | 0.69 | 0.4 | 0.52 | 0.58 |
| E | 3.0 | 1.2 | 1.2 | 0.75 | 0.0 | 2.0 | 0.55 | 1.4 | 1.26 |
| F | 1.2 | 0.75 | 0.75 | 0.69 | 2.0 | 0.0 | 0.4 | 0.82 | 0.83 |
| G | 0.67 | 1.0 | 0.76 | 0.4 | 0.55 | 0.4 | 0.0 | 0.8 | 0.57 |
| H | 2.7 | 4 | 5 | 0.52 | 1.4 | 0.82 | 0.8 | 0.0 | **1.90** |

A similar calculation can be performed for each of the nodes, and their equivalent HookeRank value is calculated, as given in Tab. 5.1. Based on the value of the HookRank score, node $H$ is elected as the top spreader in the toy network.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

### 5.2 Simulation of the proposed algorithm on real-life networks

Fig. 7, Fig. 8, and Fig. 9 depicts the final infection scale ($f(t_c)$) with respect to the percentage of spreaders for three real-life data-sets with infection rate ($\beta$) as 0.01. We consider the percentage of influential spreaders as the seed nodes in the range of 2%, 4%, 6%, 8%, and 10% to plot the final infection scale. In Fig. 7, note that the number of nodes affected by the infection is maximum for HookeRank on the US-Airports Network for most percentages of spreaders. In Fig. 8, HookeRank greatly exceeds the performance of other algorithms towards increasing the count of the spreader fraction. In the weighted PowerGrid data, shown in Fig. 9, HookeRank performs better than most other algorithms from an early stage. In the weighted PowerGrid data, shown in Fig. 10, the increase in the number of spreaders results in WVoteRank becoming marginally close to HookeRank, but our algorithm still performs better than all other algorithms in the simulation.

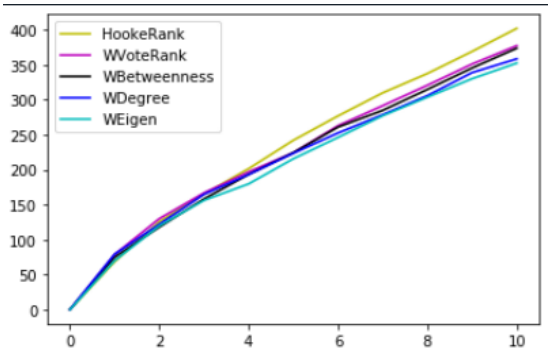


**Fig. 7.** The infection scale with respect to the percentage of spreaders on US-Airports network with $\beta = 0.01$.

Fig. 10 presents the final infection scale ($f(t_c)$) with respect to the increasing timestamps with infection rate ($\beta$) as 0.01 and top 7% influential as seed nodes on US PowerGrid network. Fig. 11 shows the final infection scale ($f(t_c)$) with respect to the increasing timestamps with infection rate ($\beta$) as 0.01 and top 5% influential as seed nodes on US PowerGrid network. Fig. 12 displays the final infection scale ($f(t_c)$) with respect to the increasing timestamps with infection rate ($\beta$) as 0.01 and top 5% influential as seed nodes on Facebook-like weighted network.

From above results on three real-life networks, it is evident that HookeRank performs better than state-of-the-art methods like weighted-degree centrality, weighted-betweenness centrality, weighted-eigenvector centrality, and weighted-voteRank, and also consistently outperforms recent methods like WVoteRank in terms of final infected scale with respect to time $t$ and spreader fraction $p$ on real-world networks as depicted in Tab. 3.3.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

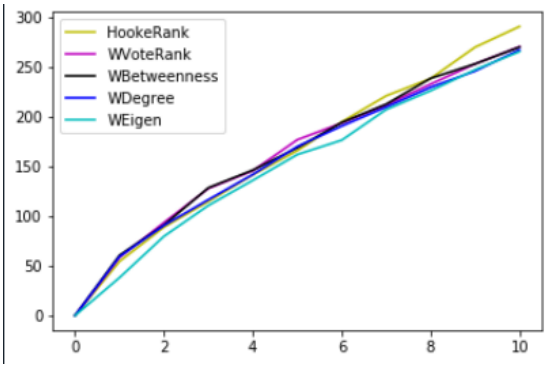


**Fig. 8.** The infection scale with respect to the percentage of spreaders on Facebook-like weighted network with $\beta = 0.01$.

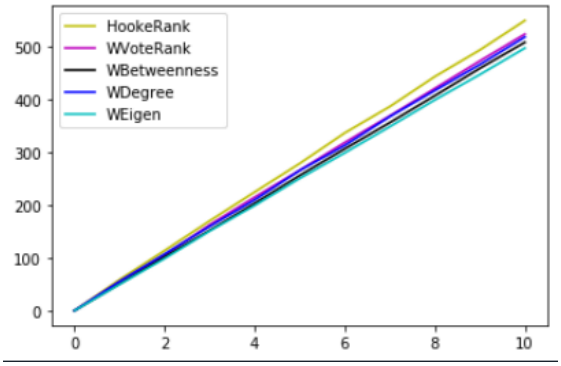


**Fig. 9.** The infection scale with respect to the percentage of spreaders on US PowerGrid network with $\beta = 0.01$.

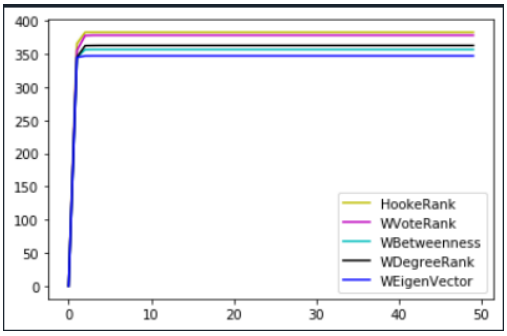


**Fig. 10.** The final infection scale with respect to the time on US PowerGrid Network with $\beta = 0.01$ and $\rho = 7\%$.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

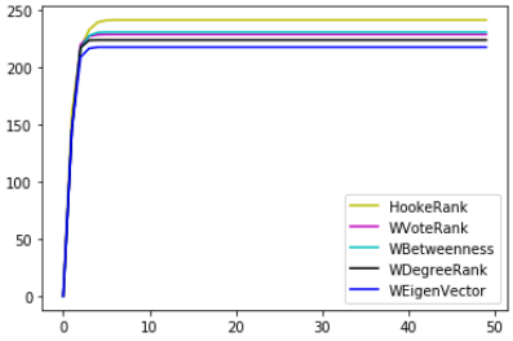


**Fig. 11.** The final infection scale with respect to the time on US-Airports Network with $\beta = 0.01$ and $\rho = 5\%$.

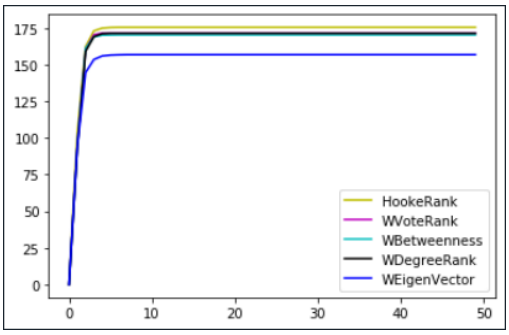


**Fig. 12.** The final infection scale with respect to the time on Facebook-like weighted Network with $\beta = 0.01$ and $\rho = 5\%$.

## 6 Conclusion

In this paper, we proposed the HookeRank method for finding influential nodes in weighted networks by modeling edges of the network as springs and edge weights as spring constants. Initially, we found a measure of the distance between indirect neighbors through the series and parallel combination of edges, by modeling them as springs. The HookeRank method and the HookeRank distance can be used to gain a better understanding of the topology in a complex weighted network. By finding the Hookerank value of the nodes, our method locates the top spreaders in the given real-world network to reach a large number of people in the network to maximize the spread of the information. The proposed algorithm incorporates both the local and global properties of a node in the measurement of its spreading capability. We performed the simulation of the proposed method along with contemporary methods on three real-life data-sets taking the basis of evaluation as the final infected scale. The proposed influence maximization algorithm performs considerably well and is effective in real-life scenarios.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

## References


1. Chen, G., Wang, X. and Li, X.: Fundamentals of complex networks: models, structures, and dynamics. John Wiley and Sons (2014).
2. Newman, M.E.: Analysis of weighted networks. Physical Review E, 70(5), p.056131(2004).
3. Valente, T.W.: Network models of the diffusion of innovations (No. 303.484 V3)(1995)
4. Kempe, D., Kleinberg, J. and Tardos, É.: Maximizing the spread of influence through a social network. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 137-146 (2003).
5. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1029-1038). ACM. (2010).
6. Wang, W., Tang, M., Stanley, H.E. and Braunstein, L.A.: Unification of theoretical approaches for epidemic spreading on complex networks. Reports on Progress in Physics, 80(3), p.036603 (2017).
7. Sun, Y., Liu, C., Zhang, C.X. and Zhang, Z.K.: Epidemic spreading on complex weighted networks. Physics Letters A, 378(7-8), pp.635-640 (2014).
8. Pastor-Satorras, R., Castellano, C., Van Mieghem, P. and Vespignani, A.: Epidemic processes in complex networks. Reviews of modern physics, 87(3), p.925 (2015).
9. Opsahl, T., Agneessens, F. and Skvoretz, J.: Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), pp.245-251 (2010).
10. Prountzos, D. and Pingali, K.: Betweenness centrality. ACM SIGPLAN Notices, 48(8), p.35.(2013).
11. Wang, H., Hernandez, J., and Van Mieghem, P.: Betweenness centrality in a weighted network (2008).
12. Zhang, J., Chen, D., Dong, Q. and Zhao, Z.: Identifying a set of influential spreaders in complex networks. Scientific Reports, 6(1) (2016).
13. Sun, H., Chen, D., He, J. and Ch'ng, E.: A voting approach to uncover multiple influential spreaders on weighted networks. Physica A: Statistical Mechanics and its Applications, 519, pp.303-312. (2019).
14. Kumar, S. and Panda, B.S.: Identifying influential nodes in Social Networks: Neighborhood Coreness based voting approach. Physica A: Statistical Mechanics and its Applications, p.124215.(2020).
15. Yu, S., Gao, L., Wang, Y.F., Gao, G., Zhou, C. and Gao, Z.Y.: Weighted H-index for identifying influential spreaders. arXiv preprint arXiv:1710.05272.(2017).
16. Bihari, A., Pandia, M. K.: Eigenvector centrality and its application in research professionals' relationship network. 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE).(2015)
17. Eades, P., Lin, X.: Spring algorithms and symmetry. Theoretical Computer Science, 240(2), 379–405. (2000).
18. Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. Software: Practice and experience, 21(11), pp.1129-1164.(1991)
19. Kamada, T. and Kawai, S.: An algorithm for drawing general undirected graphs. Information processing letters, 31(1), pp.7-15. (1989).
20. Slaughter, William S.: The Linearized Theory of Elasticity. Birkhäuser (2001).
21. Hethcote, H.W.: The mathematics of infectious diseases. SIAM Review, 42(4), pp.599-653.(2000).


ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45

22. Colizza, V., Pastor-Satorras, R., Vespignani, A.: Reaction-diffusion processes and metapopulation models in heterogeneous networks. Nature Physics 3, 276-282. (2007).
23. T. Opsahl and P. Panzarasa: Clustering in Weighted Networks, Social Networks, Vol. 31, No. 2, May, pp. 155-163. doi:10.1016/j.socnet.2009.02.002 (2009).
24. Watts, D. J., Strogatz, S. H.: Collective dynamics of "small-world" networks. Nature 393, 440-442 (1998).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_45