
DOI: [10.1371/journal.pone.0145028](http://10.1371/journal.pone.0145028)

This is the published version.

©2015, The Authors

Reproduced by Deakin University under the terms of the [Creative Commons Attribution Licence](http://creativecommons.org/licenses/by/4.0/)

Available from Deakin Research Online:

[http://hdl.handle.net/10536/DRO/DU:30081347](http://hdl.handle.net/10536/DRO/DU:30081347)
Physarum Spreader: A New Bio-Inspired Methodology for Identifying Influential Spreaders in Complex Networks

Hongping Wang¹*, Yajuan Zhang¹*, Zili Zhang¹,², Sankaran Mahadevan³, Yong Deng¹*

¹ School of Computer and Information Science, Southwest University, Chongqing, China, ² School of Information Technology, Deakin University, Geelong, VIC, Australia, ³ Civil and Environmental Engineering Department, Vanderbilt University, Nashville, TN, United States of America

☯ These authors contributed equally to this work.
* ydeng@swu.edu.cn; prof.deng@hotmail.com

Abstract

Identifying influential spreaders in networks, which contributes to optimizing the use of available resources and efficient spreading of information, is of great theoretical significance and practical value. A random-walk-based algorithm LeaderRank has been shown as an effective and efficient method in recognizing leaders in social network, which even outperforms the well-known PageRank method. As LeaderRank is initially developed for binary directed networks, further extensions should be studied in weighted networks. In this paper, a generalized algorithm Physarum Spreader is proposed by combining LeaderRank with a positive feedback mechanism inspired from an amoeboid organism called Physarum Polycephalum. By taking edge weights into consideration and adding the positive feedback mechanism, Physarum Spreader is applicable in both directed and undirected networks with weights. By taking two real networks for examples, the effectiveness of the proposed method is demonstrated by comparing with other standard centrality measures.

Introduction

Over the years, the study of graphs and networks have drawn increasing attention in a wide variety of scientific disciplines, such as biology, computer science, economics, mathematics and sociology. Meanwhile, network analysis, as a key tool to map and measure those network entities and their connections, has been well studied to provide a visual and mathematical view of networks. However, a major challenge it encounters is how to identify the most efficient spreaders for optimizing the use of available resources and ensuring efficient spread of information [1–5]. With different topology and relations, node centrality can be endowed with various meanings, such as influence [6], importance [7, 8], popularity [9, 10], controllability [11, 12] and spreading efficiency [1]. Studies on the importance of a node for spreading can be of great significance in controlling rumor, disease spreading and information flow in networks [13–16].
Ever since the idea of centrality was introduced, various centrality measures have been proposed to identify nodes which are more central than others [17–19]. Recently, Landherr et al. [20] conducted a critical review of five common centrality measures in social networks, including degree centrality [17], closeness centrality [17], betweenness centrality [17], eigenvector centrality [21] and Katz’s centrality [22]. Degree, the simplest centrality measure, is the number of edges that a node is connected to, which was firstly proposed by Freeman. However, a node with higher degree might not be in a position which can access resources quicker. In order to make up this drawback, a more sophisticated centrality measure closeness is developed, which was defined as the inverse total length from a node to all other nodes. Another important centrality measure is betweenness, which is calculated as the fraction of the times that a node lies on the shortest paths over the total number of the shortest paths. Besides, by introducing random walks, a revised betweenness centrality is proposed in [23], which counts the frequency of a node traversed by a random walk between two other nodes. And then, a random-walk-based centrality called LeaderRank [9] has been proposed, which can identify leaders in social networks better than the well-known PageRank algorithm. After that, Chen et al. [6] developed a semi-local centrality measure as a tradeoff between local degree centrality and other global but time-consuming measures. Ranking influential nodes can be seen as a multi-attribute decision making problem [24]. Due to the efficiency to combine different data [25–29], evidence theory is also widely used for identifying influential spreaders in complex networks [30, 31].

However, those measures described above are only suitable in binary networks. In many real networks, edges are with some form of attributes or weights, rather than simply either present or absent in a pair of nodes. If only binary networks are considered, ignoring the intrinsic weights attached to edges, plenty of valuable information has been lost and the analysis cannot be accurate and comprehensive. Thus, many researchers have turned their attention to centrality for weighted networks [32–36].

In 1991, Freeman et al. [37] introduced a new measure of centrality based on the concept of network flows, flow betweenness, which considered all the independent paths between all pairs of nodes in the network. Based on the degree centrality, Barrat et al. [38] proposed a measure for weighted networks, which is the sum of weights of edges that a node is connected to. Beyond that, Newman [32] and Brandes [39] have generalized the closeness and betweenness centrality for weighted networks by using Dijkstra’s algorithm [40] on computing the shortest paths. By taking both edge weights and the number of edges into consideration, a new generalization was proposed by Opsahl et al. [33], using a tuning parameter to balance the relative importance between those two parts. Later, Qi et al. [34] developed a Laplacian centrality method considering “intermediate” environmental information around a node.

In this paper, we proposed a generalized centrality metric, called PhysarumSpreader, to identify nodes with high spreading performance in networks. PhysarumSpreader is developed on the basis of a random-walk-based algorithm LeaderRank and a positive feedback mechanism inspired from an amoeboid organism, called Physarum Polycephalum. With the integration of the algorithm and mechanism, our PhysarumSpreader is applicable in both directed and undirected networks with weights. It overcomes the shortcomings of LeaderRank which is only designed for binary network and does not work well for undirected networks. Furthermore, a susceptible-infected-removed (SIR) model is employed to examine the spreading performance of nodes identified by different centrality measures. With simulations on different networks and comparison with other centrality measures, it reveals that PhysarumSpreader works well in identifying influential nodes with high spreading performance and good tolerance.

The rest of the paper is organized as follows. Section 2 begins with a brief introduction to LeaderRank algorithm and positive feedback mechanism of Physarum Polycephalum adopted in our method. Then, procedure of the proposed PhysarumSpreader for identifying influential
spreaders in networks is depicted in Section 3. And two applications in real networks are presented in section 4. What’s more, the spreading effectiveness and robustness are studied. Section 5 concludes the paper.

1 Basic Theory

1.1 LeaderRank for Identifying Leaders [9]

Given a directed network of $N$ nodes and $M$ edges, a ground node is then added by establishing bidirectional edges between it and all the other nodes, which assures the modified network as strongly connected. And the modified network consists of $N+1$ nodes and $M+2N$ directed edges. Initially, each node in the network, except for the ground node, is assigned with one unit resource, while the ground node is assigned with no resource. And then each node evenly distributes its resource to neighbors along the outgoing edges. Next is to update resource distribution as summing up the resource each node derives from its incoming edges. This process of distribution and updating of resources continues until steady state is attained. The whole process can be described mathematically as follows.

Assuming $r_i(t)$ denotes the resource of node $i$ at time $t$, the initial state ($t = 0$) of resource distribution can be represented as:

\[
\begin{align*}
  r_g(0) &= 0, \text{ where } g \text{ is ground node} \\
  r_i(0) &= 1, \text{ where } i \in \{1, 2, \ldots, N+1\} \setminus \{g\}
\end{align*}
\]  

(1)

And each node can update its resource according to the following equation:

\[
 r_i(t+1) = \sum_{j=1}^{N+1} \frac{a_{ij}}{k_{out}^j} r_j(t)
\]  

(2)

where $a_{ij}$ is the element of the corresponding ($N+1$)-dimensional adjacency matrix, which equals 1 if there is a directed link from $j$ to $i$ and 0 otherwise, and $k_{out}^j$ is the out-degree of node $j$.

When the resource $r_i(t)$ at all nodes converges to a unique steady state at time $t_c$, the resource at the ground nodes is then evenly distributed to all other nodes, and the final resource distribution on nodes $i$ is:

\[
 R_i = r_i(t_c) + \frac{r_g(t_c)}{N}, \text{ where } i \in \{1, 2, \ldots, N\}
\]  

(3)

1.2 Physarum Model for Path Finding

Physarum polycephalum, as a large, single-celled amoeboid organism, can form a dynamic tubular network within the discovered food sources.

Recently, a large amoeboid organism, Physarum polycephalum, turned out to be capable of solving many graph theoretical problems [41–45], including finding the shortest path [46–48]. Furthermore, it has been shown experimentally that the network it generates is of high intelligence and performance in road-network [49] and great transport efficiency in vascular network [50], even comparable to or better than the Tokyo rail network [51]. During the process of its path finding and tube selection, Physarum will cut off those non-competing long tubes and reinforce shorter tubes. And, with the positive feedback mechanism among the tube length, the flux through tubes and the conductivity of tubes (tube width), shorter tubes result in larger flux; tube with a large flux grow (tube width increases); wider tubes leads to a further increase
of flux as the resistance to the flow decreases in wider tubes. According to Physarum’s tubular network, each tube segment is regarded as an edge $e_{ij}$ in graph and its two ends are denoted as nodes $i$ and $j$, which the edge connects. For each tube segment, there are two critical attributes: one is the length of the tube $L_{ij}$ and the other is its thickness, which is always represented as conductivity $D_{ij}$. Based on the theory that thick, short tubes are typically the most effective for transportation, the mathematical model of Physarum includes two parts: flux through tubes and adaptation of conductivity according to its flux.

### 1.2.1 Flux through Tubes [52]

As circulation is based on streaming through network of tubular channels, the flux of sol through the tubes $Q_{ij}$ is approximately modeled as Poiseuille flow:

\[
Q_{ij} = \frac{D_{ij} L_{ij}}{\pi} \left( p_i - p_j \right)
\]

(4)

where $p_i$ is the pressure at node $i$.

By considering the balance of flux through each node, we have

\[
\sum \frac{D_{ij}}{L_{ij}} (p_i - p_j) = \begin{cases} 
-L_0, & \text{for } j = s, \\
+L_0, & \text{for } j = t, \\
0, & \text{otherwise.}
\end{cases}
\]

(5)

where $s$ is the source node that the initial flux $I_0$ flows out, while $t$ is the sink node from which the flux flows in.

### 1.2.2 Adaptation of Conductivity [52]

In order to model the positive feedback mechanism that tube widens with increasing flux and degenerates with decreasing flux, the conductivity $D_{ij}$ is assumed to change over time according to the flux $Q_{ij}$:

\[
\frac{d}{dt} D_{ij} = f(|Q_{ij}|) - \gamma D_{ij}
\]

(6)

where $\gamma$ is a decay rate of the tube. $f(Q)$ is an increasing function with $f(0) = 0$. More detailed description of $f(Q)$ can be found in [53].

## 2 PhysarumSpreader

LeaderRank is an efficient method for identifying influential leaders in opinion spreading and outperforms PageRank algorithm, the basis of the Google search engine, in ranking effectiveness and robustness against manipulations and noisy data. However, it is initially designed for binary networks, which is not suitable for weighted networks.

Consider that the resource distribution from each node to its neighbors in LeaderRank, is similar to the flowing of flux through tubes in Physarum model of path finding. Nevertheless, Physarum model is designed for finding the shortest paths in both binary and weighted networks, which is capable of handling edge weights. Thus, it is natural to consider that adoption of the positive feedback mechanism between conductivity and flux in Physarum model may be of great help in overcoming the weakness of LeaderRank in weighted networks.

The main mechanism behind the presented method is combining the positive feedback mechanism in Physarum model and resources distribution mechanism in LeaderRank. Specifically, each node proportionally distributes resources to its outlinks based on their weights and conductivities. Then, a positive feedback mechanism is employed to accelerate convergence of the algorithm. Here, the positive feedback mechanism is an interaction between conductivities and resources along each link. A link with few resources leads to a weak conductivities. The...
weak conductivities produce a further decrease of resources along the link. Similarly, a link with more resources cases stronger conductivities and further contributes itself to obtain more resources. Finally, the resources of a node are the sum of its inlinks’ resources. This process will continue until each nodes’ resources are steady.

Therefore, in this paper, based on the primary LeaderRank, an extended algorithm, called PhysarumSpreader, is proposed for capturing the spreading ability of nodes in weighted networks.

2.1 General Flow of PhysarumSpreader

The general flow of our proposed PhysarumSpreader is described as follows, along with graphical demonstration of a random directed example network Net as shown in Fig 1.

**Step 1** Add a ground node into the network by connecting every node through bidirectional links (Fig 2). The weight $w_{ig}$ of the inlink which direction is from node $i(i \in N)$ to ground node is determined by the following equation:

$$w_{ig} = \begin{cases} \frac{\sum_{j=1}^{n} w_{ji}}{l_{i}^{out}}, & \text{if } l_{i}^{out} > 0 \\ 1, & \text{if } l_{i}^{out} = 0 \end{cases} \quad (7)$$

![Fig 1. An example network Net.](https://example.com/fig1.png)

doi:10.1371/journal.pone.0145028.g001
where $l_{i}^{\text{out}}$ denotes the total number of outlinks from node $i$ without considering weight and node $j$ represents the neighbour of node $i$. If the network is directed, this step guarantees the network to be strongly connected.

**Step 2** Initialize all nodes (other than the ground node) with unit of resource and the ground node with a score of 0 (Fig 3).

\[
I_g(0) = 0, \text{ where } g \text{ is ground node}
\]

\[
I_i(0) = 1, \text{ where } i \in \{1, 2, \cdots, N + 1\} \setminus \{g\}
\]  

**Step 3** Distribute each node’s flux to its neighbors through the out-going edges according to their edge weights.

\[
Q_i(t) = \frac{w_i D_i(t)}{N+1} \sum_{j=1}^{N+1} w_j D_j(t)
\]
where $D_{ij}$ is the conductivity of each edge with initial value as 1. It is notable that the value of $w_{ij}$ varies under different circumstances. If the given network is binary, $w_{ij} = 1$ for all edges in the network. If the network is weighted and the weight refers to the cost of traversing the edge, $w_{ij}$ will be the reciprocal of the edge weight. But if the weight stands for the strength of edge relation, such as the number of social proximities, $w_{ij}$ will be assigned as the weight. The initial flux distribution on each out-going edge of nodes at time $t = 0$ is calculated, shown in matrix $Q(0)$:

$$Q(0) = \begin{bmatrix}
0 & 0.5 & 0 & 0 & 0.5 \\
0 & 0 & 0.5 & 0 & 0.5 \\
0.5 & 0 & 0 & 0 & 0.5 \\
0.33 & 0.33 & 0 & 0 & 0.33 \\
0.25 & 0.25 & 0.25 & 0.25 & 0
\end{bmatrix}$$
Step 4 Adapt the conductivity according the flux through each edge using the following equation:

$$D_{ij}(t+1) = \left[ Q_{ij}(t) + D_{ij}(t) \right]/2, \text{ where } i, j \neq g$$

(10)

The edge conductivity at time $t=1$ can be calculated by Eq 10, according to the flux distribution of time $t=0$. The result is shown in matrix $D(1)$:

$$D(1) = 
\begin{bmatrix}
0 & 0.75 & 0 & 0 & 1 \\
0 & 0 & 0.75 & 0 & 1 \\
0.75 & 0 & 0 & 0 & 1 \\
0.67 & 0.67 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}$$

Step 5 Update the resources of each node for the next iteration, according to the flux flowing into the node and its current score.

$$I_i(t+1) = \sum_{j=1}^{N+1} Q_{ji}(t) I_j(t)$$

(11)

Step 6 Determine whether the steady state of nodes’ score is attained. If it converges to a steady state, the flux of the ground node is evenly distributed to all other nodes. And the final score $S_i$ for each node’s spreading performance is attained as:

$$S_i = I_i(t_c) + \frac{I_i(t_c)}{N}, \text{ where } i \in \{1, 2, \cdots, N\}$$

(12)

If the state is not steady yet, the process continues to Step 3.

The final resources for each node in Net is: $S_1(1.0100) > S_2(1.0097) > S_3(1.0002) > S_4(0.9801)$.

3 Comparisons and Tests

In this section, the proposed method is compared with another four approaches (degree, betweenness, k-shell, weighted PageRank) to demonstrate its effectiveness. The four methods will be defined in section 3.1. What’s more, all methods are tested on noisy data to evaluate their stability. Two real networks are employed. One of them is a directed and weighted network. It is the network of the 500 busiest commercial airports in the United States. A tie exists between two airports if a flight was scheduled between them in 2002. The weights correspond to the number of seats available on the scheduled flights [54]. You can obtain the data through the hyperlink listed in S1 Text. The other one is an undirected and weighted network reorganized by Newman [55–57]. It is a collaboration network of scientists posting preprints on the high-energy theory archive at www.arxiv.org. These papers appeared in a 5-year window, from 1995 to 1999 inclusive. The data can also be downloaded through the S1 Text. More attributes about the two networks are listed in Table 1.
3.1 Definitions of the compared centrality measures

In the context of social science, the topology of a social network is represented by an adjacency matrix $A = \{w_{ij}\}_{N \times N}$, where the element $w_{ij} > 0$ if there exists a link from $j$ to $i$ and $w_{ij} = 0$ otherwise. For an undirected network, $A$ is a symmetric matrix with $w_{ij} = w_{ji}$. If the network is weighted, the element $w_{ij}$ represents the weight of the link from $j$ to $i$. Actually, the adjacency matrix $A$ fully describes the topological structure of the social network. Here, the adopted centrality measures to make comparisons are calculated by the following equations:

(1) Degree $k_i$ for a node $i$ can be computed as follows:

$$k_i = \sum_{j=1}^{N} w_{ij} + \sum_{j=1}^{N} w_{ji}$$

(13)

(2) Betweenness $C_B(i)$ is defined as

$$C_B(i) = \sum_{s \neq i \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

(14)

where $\sigma_{st}$ is the number of the shortest paths between nodes $s$ and $t$, and $\sigma_{st}(i)$ is the number of the shortest paths between $s$ and $t$ which pass through node $i$.

(3) K-shell [58]: The $k$-shell index of a node is obtained by a procedure called $k$-shell decomposition, where we successively prune nodes in the network layer by layer. Concretely, the decomposition starts by removing nodes with degree $k = 1$. After that, some nodes may have only one link left. So we continue pruning the network iteratively until there are no nodes with $k = 1$. The removed nodes fall into a $k$-shell with index $k_S = 1$. With the similar method, we iteratively remove the next $k$ shell $k_S = 2$ and higher $k$ shells until all nodes are pruned. In the decomposition procedure, each node is assigned with a $k$-shell index. The periphery of the network corresponds to small $k_S$ and the nodes with high $k_S$ define the core of the network.

(4) Weighted PageRank [59] can be calculated from:

$$PR_i(t) = \alpha \sum_{j=1}^{N} \frac{w_{ij} PR_j(t)}{k_i^{out}} + \frac{1 - \alpha}{N}$$

(15)

where $k_i^{out} = \sum_{i=1}^{N} w_{ij}$ and $\alpha$ is the jumping probability. $PR_i(t)$ is the probability that node $i$ is visited by the random walker at time $t$. As time $t$ increases, the probability $PR_i(t)$ will converge to a stationary probability $PR_i$. This value is defined as the PageRank which are used to determine its ranking relative to other nodes. In the calculation, the conventional choice of $\alpha$ is 0.85. In this paper, $\alpha$ is set as 0.85 for all experiments.

3.2 Effectiveness

A modified susceptible-infected-removed (SIR) model is employed to estimate the spreading influence of the top-ranked nodes in weighted networks. In this model, individuals can be in three discrete states: susceptible, infected or removed. Each individual in the model can be
represented by a node of the network and can only spread infection to its neighbors along the outgoing edges in the network. At each step, each infected node $i$ randomly chooses one of its susceptible neighbors, $j$, and infects it with probability $\lambda_{ij}$, and then be removed (dead or recovered with immunity) with probability $\beta$. The probability $\lambda_{ij}$ is determined by the following equation [60]:

$$\lambda_{ij} = \left( \frac{w_{ij}}{w_{\text{max}}} \right)^\alpha, \quad \alpha > 0$$

where $\alpha$ is a positive constant and $w_{\text{max}}$ is the largest value of $w_{ij}$ in the network. Since $\frac{w_{ij}}{w_{\text{max}}} < 1$, the smaller the $\alpha$ is, the more quickly the infection spreads. The process stops when no infected node is present. Here we use the cumulative number of infected nodes (which includes infected and recover nodes), denoted by $N$, as a function time. Without the loss of generality, $\alpha$ and $\beta$ are assigned as 0.2 and 1.

We first compare the spreading processed activated by top-ranked [61] nodes from PhysarumSpreader and another four centrality measures (degree, betweenness, k-shell, weighted PageRank). Taking weighted PageRank as an example. If among two top-$L$ lists by PhysarumSpreader and weighted PageRank, there are $n$ different nodes, we compare the spreading processed activated by these $n$ different nodes, respectively. For example, if $L = 5$, the top-ranked lists are $\{97, 1016, 754, 333, 335\}$ and $\{754, 1222, 97, 841, 1016\}$ for weighted PageRank and PhysarumSpreader, then $n = 3$ and the spreading processes are compared with initially infected nodes $\{333, 335\}$ for weighted PageRank and $\{1222, 841\}$ for PhysarumSpreader.

Tables 2 and 3 list the top 50 nodes obtained by different centrality methods in US airports network and collaborations network.

In US airports network, Fig 4a, 4b, 4c and 4d show the spreading results compared with degree centrality, betweenness, k-shell and weighted PageRank corresponding to $n = 11, 17, 16$ and 10, respectively, when $L$ equals to 20. As can be seen, the proposed method slightly outperforms the other four centrality measures. In order to verify the efficacy of the proposed method, we check its efficiency among more nodes. Since it is impossible to check all nodes, we selected the 50 most important nodes to conduct the experiment. Fig 5a, 5b, 5c and 5d display the spreading results corresponding to $n = 24, 35, 27$ and 18 under $L = 50$. Here we can see, the proposed algorithm exhibits a good efficiency in $L = 50$ as well as in $L = 20$.

We also applied our algorithm in the other network with bigger size: collaborations network. Fig 6a, 6b, 6c and 6d show the spreading results compared with another four centrality measures corresponding to $n = 11, 16, 20$ and 10, respectively, under $L = 20$. The figures show that the cumulative number of the infected nodes obtained by our method is a bit bigger than the results calculated by the others except k-shell. However, when $L$ equals to 50 in Fig 7, our method is a little bit worse than other centrality measures but has a quicker spread velocity than k-shell while $n = 29, 43, 47$ and 27.

With $L$ increasing, more and more important nodes overlap, those overlapped nodes will be removed from the top-ranked lists. It means that the rest different nodes may have less spreading influence than those removed. Thus, the cumulative numbers of infected nodes have little changes in spite of $n$ and $L$ increasing.

### 3.3 Robustness

To scientifically test the performance of the PhysarumSpreader algorithm, we measure the change in rankings when links are randomly removed with probability form 0.1 to 0.5. The rankings obtained from the modified network are compared to those from the original
Table 2. Top 50 nodes ranked by different centrality methods for US airports network.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Degree</th>
<th>Betweenness</th>
<th>K-shell</th>
<th>Weighted PageRank</th>
<th>PhysarumSpreader</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>97</td>
<td>76</td>
<td>27</td>
<td>97</td>
<td>754</td>
</tr>
<tr>
<td>2</td>
<td>1016</td>
<td>584</td>
<td>65</td>
<td>1016</td>
<td>1222</td>
</tr>
<tr>
<td>3</td>
<td>604</td>
<td>1331</td>
<td>97</td>
<td>754</td>
<td>97</td>
</tr>
<tr>
<td>4</td>
<td>654</td>
<td>1433</td>
<td>106</td>
<td>333</td>
<td>841</td>
</tr>
<tr>
<td>5</td>
<td>754</td>
<td>207</td>
<td>126</td>
<td>335</td>
<td>1016</td>
</tr>
<tr>
<td>6</td>
<td>422</td>
<td>591</td>
<td>148</td>
<td>654</td>
<td>774</td>
</tr>
<tr>
<td>7</td>
<td>333</td>
<td>604</td>
<td>176</td>
<td>1219</td>
<td>1219</td>
</tr>
<tr>
<td>8</td>
<td>907</td>
<td>428</td>
<td>181</td>
<td>76</td>
<td>76</td>
</tr>
<tr>
<td>9</td>
<td>606</td>
<td>892</td>
<td>184</td>
<td>606</td>
<td>333</td>
</tr>
<tr>
<td>10</td>
<td>335</td>
<td>1520</td>
<td>206</td>
<td>1222</td>
<td>654</td>
</tr>
<tr>
<td>11</td>
<td>866</td>
<td>126</td>
<td>207</td>
<td>866</td>
<td>1544</td>
</tr>
<tr>
<td>12</td>
<td>368</td>
<td>1542</td>
<td>212</td>
<td>752</td>
<td>790</td>
</tr>
<tr>
<td>13</td>
<td>1062</td>
<td>1129</td>
<td>261</td>
<td>1064</td>
<td>33</td>
</tr>
<tr>
<td>14</td>
<td>752</td>
<td>654</td>
<td>267</td>
<td>907</td>
<td>325</td>
</tr>
<tr>
<td>15</td>
<td>184</td>
<td>343</td>
<td>271</td>
<td>267</td>
<td>582</td>
</tr>
<tr>
<td>16</td>
<td>841</td>
<td>1557</td>
<td>306</td>
<td>422</td>
<td>599</td>
</tr>
<tr>
<td>17</td>
<td>445</td>
<td>1006</td>
<td>325</td>
<td>841</td>
<td>985</td>
</tr>
<tr>
<td>18</td>
<td>1222</td>
<td>427</td>
<td>328</td>
<td>368</td>
<td>207</td>
</tr>
<tr>
<td>19</td>
<td>267</td>
<td>686</td>
<td>333</td>
<td>1062</td>
<td>1457</td>
</tr>
<tr>
<td>20</td>
<td>1219</td>
<td>136</td>
<td>335</td>
<td>184</td>
<td>866</td>
</tr>
<tr>
<td>21</td>
<td>207</td>
<td>431</td>
<td>368</td>
<td>604</td>
<td>132</td>
</tr>
<tr>
<td>22</td>
<td>261</td>
<td>60</td>
<td>405</td>
<td>1250</td>
<td>241</td>
</tr>
<tr>
<td>23</td>
<td>212</td>
<td>754</td>
<td>422</td>
<td>445</td>
<td>427</td>
</tr>
<tr>
<td>24</td>
<td>176</td>
<td>1238</td>
<td>445</td>
<td>573</td>
<td>335</td>
</tr>
<tr>
<td>25</td>
<td>850</td>
<td>166</td>
<td>497</td>
<td>774</td>
<td>907</td>
</tr>
<tr>
<td>26</td>
<td>1557</td>
<td>206</td>
<td>582</td>
<td>212</td>
<td>606</td>
</tr>
<tr>
<td>27</td>
<td>1064</td>
<td>866</td>
<td>604</td>
<td>427</td>
<td>573</td>
</tr>
<tr>
<td>28</td>
<td>1072</td>
<td>722</td>
<td>606</td>
<td>1190</td>
<td>1250</td>
</tr>
<tr>
<td>29</td>
<td>1250</td>
<td>1051</td>
<td>624</td>
<td>1050</td>
<td>136</td>
</tr>
<tr>
<td>30</td>
<td>998</td>
<td>93</td>
<td>649</td>
<td>328</td>
<td>264</td>
</tr>
<tr>
<td>31</td>
<td>584</td>
<td>1287</td>
<td>654</td>
<td>132</td>
<td>1006</td>
</tr>
<tr>
<td>32</td>
<td>870</td>
<td>1032</td>
<td>752</td>
<td>846</td>
<td>267</td>
</tr>
<tr>
<td>33</td>
<td>1148</td>
<td>1566</td>
<td>754</td>
<td>1369</td>
<td>1064</td>
</tr>
<tr>
<td>34</td>
<td>838</td>
<td>1263</td>
<td>774</td>
<td>790</td>
<td>60</td>
</tr>
<tr>
<td>35</td>
<td>328</td>
<td>573</td>
<td>787</td>
<td>1287</td>
<td>422</td>
</tr>
<tr>
<td>36</td>
<td>774</td>
<td>107</td>
<td>838</td>
<td>779</td>
<td>368</td>
</tr>
<tr>
<td>37</td>
<td>624</td>
<td>394</td>
<td>841</td>
<td>850</td>
<td>1029</td>
</tr>
<tr>
<td>38</td>
<td>1287</td>
<td>365</td>
<td>846</td>
<td>1557</td>
<td>1426</td>
</tr>
<tr>
<td>39</td>
<td>271</td>
<td>1125</td>
<td>850</td>
<td>1457</td>
<td>78</td>
</tr>
<tr>
<td>40</td>
<td>106</td>
<td>466</td>
<td>863</td>
<td>1244</td>
<td>492</td>
</tr>
<tr>
<td>41</td>
<td>910</td>
<td>1294</td>
<td>866</td>
<td>985</td>
<td>184</td>
</tr>
<tr>
<td>42</td>
<td>846</td>
<td>422</td>
<td>870</td>
<td>968</td>
<td>661</td>
</tr>
<tr>
<td>43</td>
<td>1212</td>
<td>907</td>
<td>907</td>
<td>584</td>
<td>39</td>
</tr>
<tr>
<td>44</td>
<td>614</td>
<td>498</td>
<td>910</td>
<td>838</td>
<td>604</td>
</tr>
<tr>
<td>45</td>
<td>1192</td>
<td>184</td>
<td>985</td>
<td>870</td>
<td>968</td>
</tr>
<tr>
<td>46</td>
<td>148</td>
<td>670</td>
<td>998</td>
<td>261</td>
<td>445</td>
</tr>
<tr>
<td>47</td>
<td>76</td>
<td>752</td>
<td>1005</td>
<td>1258</td>
<td>1062</td>
</tr>
</tbody>
</table>

(Continued)
Table 2. (Continued)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Degree</th>
<th>Betweenness</th>
<th>K-shell</th>
<th>Weighted PageRank</th>
<th>PhysarumSpreader</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>306</td>
<td>710</td>
<td>1008</td>
<td>1239</td>
<td>752</td>
</tr>
<tr>
<td>49</td>
<td>1369</td>
<td>503</td>
<td>1016</td>
<td>1263</td>
<td>212</td>
</tr>
<tr>
<td>50</td>
<td>126</td>
<td>36</td>
<td>1017</td>
<td>1148</td>
<td>342</td>
</tr>
</tbody>
</table>

doi:10.1371/journal.pone.0145028.t002

Table 3. Top 50 nodes ranked by different centrality methods for collaborations network.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Degree</th>
<th>Betweenness</th>
<th>K-shell</th>
<th>Weighted PageRank</th>
<th>PhysarumSpreader</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>87</td>
<td>1571</td>
<td>6790</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>480</td>
<td>1832</td>
<td>6791</td>
<td>87</td>
<td>1794</td>
</tr>
<tr>
<td>3</td>
<td>168</td>
<td>24</td>
<td>6792</td>
<td>763</td>
<td>547</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>975</td>
<td>6793</td>
<td>997</td>
<td>546</td>
</tr>
<tr>
<td>5</td>
<td>997</td>
<td>3473</td>
<td>6794</td>
<td>546</td>
<td>473</td>
</tr>
<tr>
<td>6</td>
<td>481</td>
<td>500</td>
<td>6795</td>
<td>530</td>
<td>415</td>
</tr>
<tr>
<td>7</td>
<td>656</td>
<td>1603</td>
<td>6796</td>
<td>547</td>
<td>763</td>
</tr>
<tr>
<td>8</td>
<td>926</td>
<td>1368</td>
<td>6797</td>
<td>480</td>
<td>474</td>
</tr>
<tr>
<td>9</td>
<td>1571</td>
<td>926</td>
<td>6798</td>
<td>168</td>
<td>106</td>
</tr>
<tr>
<td>10</td>
<td>546</td>
<td>87</td>
<td>6799</td>
<td>469</td>
<td>926</td>
</tr>
<tr>
<td>11</td>
<td>975</td>
<td>1617</td>
<td>6800</td>
<td>433</td>
<td>123</td>
</tr>
<tr>
<td>12</td>
<td>473</td>
<td>276</td>
<td>6801</td>
<td>1424</td>
<td>2855</td>
</tr>
<tr>
<td>13</td>
<td>530</td>
<td>1337</td>
<td>6802</td>
<td>1201</td>
<td>87</td>
</tr>
<tr>
<td>14</td>
<td>547</td>
<td>481</td>
<td>6803</td>
<td>1794</td>
<td>480</td>
</tr>
<tr>
<td>15</td>
<td>884</td>
<td>193</td>
<td>6804</td>
<td>675</td>
<td>1340</td>
</tr>
<tr>
<td>16</td>
<td>930</td>
<td>930</td>
<td>6805</td>
<td>816</td>
<td>469</td>
</tr>
<tr>
<td>17</td>
<td>1603</td>
<td>2544</td>
<td>6806</td>
<td>975</td>
<td>39</td>
</tr>
<tr>
<td>18</td>
<td>39</td>
<td>480</td>
<td>6807</td>
<td>926</td>
<td>124</td>
</tr>
<tr>
<td>19</td>
<td>123</td>
<td>6487</td>
<td>6808</td>
<td>2117</td>
<td>1452</td>
</tr>
<tr>
<td>20</td>
<td>956</td>
<td>387</td>
<td>6809</td>
<td>106</td>
<td>479</td>
</tr>
<tr>
<td>21</td>
<td>106</td>
<td>168</td>
<td>6810</td>
<td>415</td>
<td>3940</td>
</tr>
<tr>
<td>22</td>
<td>675</td>
<td>752</td>
<td>6811</td>
<td>164</td>
<td>1087</td>
</tr>
<tr>
<td>23</td>
<td>1171</td>
<td>165</td>
<td>6812</td>
<td>2254</td>
<td>197</td>
</tr>
<tr>
<td>24</td>
<td>1201</td>
<td>307</td>
<td>6813</td>
<td>2855</td>
<td>38</td>
</tr>
<tr>
<td>25</td>
<td>1516</td>
<td>4201</td>
<td>310</td>
<td>177</td>
<td>420</td>
</tr>
<tr>
<td>26</td>
<td>2117</td>
<td>1190</td>
<td>5687</td>
<td>1941</td>
<td>3402</td>
</tr>
<tr>
<td>27</td>
<td>38</td>
<td>5248</td>
<td>5688</td>
<td>1698</td>
<td>1548</td>
</tr>
<tr>
<td>28</td>
<td>310</td>
<td>502</td>
<td>5689</td>
<td>884</td>
<td>431</td>
</tr>
<tr>
<td>29</td>
<td>763</td>
<td>1738</td>
<td>5690</td>
<td>1754</td>
<td>1754</td>
</tr>
<tr>
<td>30</td>
<td>1393</td>
<td>2286</td>
<td>5691</td>
<td>896</td>
<td>195</td>
</tr>
<tr>
<td>31</td>
<td>2070</td>
<td>5908</td>
<td>5692</td>
<td>1528</td>
<td>1641</td>
</tr>
<tr>
<td>32</td>
<td>1286</td>
<td>4490</td>
<td>5693</td>
<td>1479</td>
<td>37</td>
</tr>
<tr>
<td>33</td>
<td>185</td>
<td>1275</td>
<td>5694</td>
<td>473</td>
<td>1511</td>
</tr>
<tr>
<td>34</td>
<td>193</td>
<td>1515</td>
<td>5695</td>
<td>474</td>
<td>1870</td>
</tr>
<tr>
<td>35</td>
<td>431</td>
<td>6505</td>
<td>5696</td>
<td>1444</td>
<td>1512</td>
</tr>
<tr>
<td>36</td>
<td>433</td>
<td>3140</td>
<td>5697</td>
<td>336</td>
<td>178</td>
</tr>
<tr>
<td>37</td>
<td>469</td>
<td>997</td>
<td>5698</td>
<td>38</td>
<td>517</td>
</tr>
<tr>
<td>38</td>
<td>4201</td>
<td>185</td>
<td>5699</td>
<td>41</td>
<td>1571</td>
</tr>
</tbody>
</table>

(Continued)
network, by measuring the impact $I_R$ on ranking, as given by [9]

$$I_R = \sum_{i=1}^{N} |R'_i - R_i|$$ (16)

Table 3. (Continued)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Degree</th>
<th>Betweenness</th>
<th>K-shell</th>
<th>Weighted PageRank</th>
<th>PhysarumSpreader</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>221</td>
<td>890</td>
<td>5700</td>
<td>1736</td>
<td>1642</td>
</tr>
<tr>
<td>40</td>
<td>449</td>
<td>166</td>
<td>5701</td>
<td>1452</td>
<td>518</td>
</tr>
<tr>
<td>41</td>
<td>1126</td>
<td>1393</td>
<td>5702</td>
<td>39</td>
<td>530</td>
</tr>
<tr>
<td>42</td>
<td>1255</td>
<td>1479</td>
<td>5703</td>
<td>23</td>
<td>322</td>
</tr>
<tr>
<td>43</td>
<td>1479</td>
<td>3770</td>
<td>5704</td>
<td>1854</td>
<td>2116</td>
</tr>
<tr>
<td>44</td>
<td>1570</td>
<td>404</td>
<td>53</td>
<td>85</td>
<td>2217</td>
</tr>
<tr>
<td>45</td>
<td>1698</td>
<td>501</td>
<td>530</td>
<td>956</td>
<td>3717</td>
</tr>
<tr>
<td>46</td>
<td>1754</td>
<td>948</td>
<td>546</td>
<td>616</td>
<td>321</td>
</tr>
<tr>
<td>47</td>
<td>35</td>
<td>1058</td>
<td>547</td>
<td>631</td>
<td>1640</td>
</tr>
<tr>
<td>48</td>
<td>41</td>
<td>956</td>
<td>1831</td>
<td>592</td>
<td>433</td>
</tr>
<tr>
<td>49</td>
<td>371</td>
<td>103</td>
<td>4836</td>
<td>1241</td>
<td>956</td>
</tr>
<tr>
<td>50</td>
<td>474</td>
<td>1247</td>
<td>5248</td>
<td>420</td>
<td>168</td>
</tr>
</tbody>
</table>

doi:10.1371/journal.pone.0145028.t003

doi:10.1371/journal.pone.0145028.g004

Fig 4. Comparison of epidemic spreading efficiency among different measures when $L = 20$, in US airports network. Each result $N$ is obtained by averaging over 100 implementations with $\alpha = 0.2$ and $\beta = 1$. (a):PhysarumSpreader vs Degree, $n = 11$. (b):PhysarumSpreader vs Betweenness, $n = 17$. (c): PhysarumSpreader vs K-shell, $n = 16$. (d):PhysarumSpreader vs Weighted PageRank, $n = 10$. 

doi:10.1371/journal.pone.0145028.g004
3 and $R_i$ correspond to the rankings obtained respectively from the original and modified graph. We measure $I_k$ for PhysarumSpreader, weighted PageRank, degree, betweenness and k-shell subject to the same modifications.

As shown in Figs 8 and 9, $I_k$ increase with the number of links removed. In Fig 8, we can observe that our method is obviously more tolerant than weighted PageRank, betweenness and k-shell but worse than degree.

Here we can see, robustness of the presented approach is also better than weighted PageRank, in Fig 9. However, when $p > 0.2$, the tolerance of our method underperform degree and is a little worse than k-shell while $p > 0.3$. Notably, it seems that betweenness becomes the most tolerant method, but it’s not true. Due to the topology of the original network, there are 4568 nodes whose value of betweenness equal to 0. It means that the rankings of these nodes remain unchanged when the network is modified. Hence, the tolerance of betweenness is evaluated invalidly, when it is tested in collaborations network by this measure.

In practical application, a valuable ranking algorithm should not be effective only, but also be robust. To further demonstrate the proposed method’s advantages in terms of robustness, we demonstrate PhysarumSpreader algorithm in a representative example called sybil attack [62]. Consider a situation that spammers deliberately gain disproportionately high rank by creating huge fake entities. To simulate this attack, each time node $i$ creates $v$ ($v = 10, 50, 100$) fake entities which only direct to node $i$ with weight equaling to 1. The rank of node $i$ is denoted by

![Fig 5. Comparison of epidemic spreading efficiency among different measures when $L = 50$, in US airports network.](image)

Each result $N$ is obtained by averaging over 100 implementations with $a = 0.2$ and $\beta = 1$. (a): PhysarumSpreader vs Degree, $n = 24$. (b): PhysarumSpreader vs Betweenness, $n = 35$. (c): PhysarumSpreader vs K-shell, $n = 27$. (d): PhysarumSpreader vs Weighted PageRank, $n = 18$. 

doi:10.1371/journal.pone.0145028.g005
\( r_i \) first and then it is represented by \( r'_i \) after manipulations. Obviously, \( r'_i \) is less than or equal to \( r_i \) and if there are smaller differences between them, the approach will be more robust. Only the top-100 users \((i = 1, 2, 3, \ldots, 100)\) are studied under this attack.

As shown in Figs 10 and 11, the vertical axis displays the manipulated rank of a user after creation of \( v \) fake fans and the horizontal axis shows its original rank. As we can see, in both networks of US airports network and collaborations network, \( \text{Physarum Spreader} \) is the most robust algorithm among others against manipulations as its smaller change of rank.

4 Conclusions

In this paper, focus is placed on identifying influential spreaders in weighted networks and an extended algorithm called \( \text{Physarum Spreader} \) has been proposed based on LeaderRank and a positive feedback mechanism inspired from \( \text{Physarum Polycephalum} \). In order to investigate the performance of the proposed method, two weighted real networks that one is directed and the other is undirected, have been used as test network data sets. Furthermore, comparison with four well-known centrality measures are also studied with the help of an epidemic spreading model. Experimental results indicate that \( \text{Physarum Spreader} \) is effective in identifying influential spreaders. In addition, the proposed method has a good robustness compared with other measures.
Fig 7. Comparison of epidemic spreading efficiency among different measures when $L = 50$, in collaborations network. Each result $N$ is obtained by averaging over 100 implementations with $\alpha = 0.2$ and $\beta = 1$. (a): Physarum Spreader vs Degree, $n = 29$. (b): Physarum Spreader vs Betweenness, $n = 43$. (c): Physarum Spreader vs K-shell, $n = 47$. (d): Physarum Spreader vs Weighted PageRank, $n = 27$.

doi:10.1371/journal.pone.0145028.g007

Fig 8. The impact ranking as a function of number of links randomly removed for US airports network. All data points are average values over 100 independent runs with error bars showing the standard deviations.

doi:10.1371/journal.pone.0145028.g008
Fig 9. The impact ranking as a function of number of links randomly removed for collaborations network. All data points are average values over 100 independent runs with error bars showing the standard deviations.

doi:10.1371/journal.pone.0145028.g009

Fig 10. The impact ranking as a function of number of links randomly removed for US airports network. All data points are average values over 100 independent runs with error bars showing the standard deviations.

doi:10.1371/journal.pone.0145028.g010
Fig 11. The impact ranking as a function of number of links randomly removed for collaborations network. All data points are average values over 100 independent runs with error bars showing the standard deviations.

doi:10.1371/journal.pone.0145028.g011

Supporting Information
S1 Text. Dataset web pages.

(TXT)

Acknowledgments
The authors greatly appreciate the reviews’ suggestions and the editor’s encouragement. The work is partially supported by National High Technology Research and Development Program of China (863 Program) (Grant No. 2013AA013801), National Natural Science Foundation of China (Grant Nos. 61174022, 61573290, 61503237), China State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No. BUAA-VR-14KF-02).

Author Contributions
Conceived and designed the experiments: HW YZ YD. Performed the experiments: YZ ZZ. Analyzed the data: HW YD SM. Wrote the paper: YZ HW.

References


