

## ORIGINAL ARTICLE

# Identifying Influential Nodes Using a Shell-Based Ranking and Filtering Method in Social Networks

Hamid Ahmadi Beni and Asgarali Bouyer\*

### Abstract

The main goal in the influence maximization problem (IMP) is to find  $k$  minimum nodes with the highest influence spread on the social networks. Since IMP is NP-hard and is not possible to obtain the optimum results, it is considered by heuristic algorithms. Many strategies focus on the power of the influence spread of core nodes to find  $k$  influential nodes. Most of the core detection-based methods concentrate on nodes in the highest core and often give the same power for all nodes in the best core. However, some other nodes fairly have the potential to select as seed nodes in other less important cores, because these nodes can play an important role in the diffusion of information among the core nodes with other nodes. Given this fact, this article proposes a new shell-based ranking and filtering method, called shell-based ranking and filtering method (SRFM), for selecting influential seeds with the aim to maximize influence in the network. The proposed algorithm initially selects a set of nodes in different shells. Moreover, a set of the candidate nodes are created, and most of the periphery nodes are removed during a pruning approach to reduce the computational overhead. Therefore, the seed nodes are selected from the candidate nodes set using the role of the bridge nodes. Experimental results in both synthetic and real data sets showed that the proposed SRFM algorithm has more acceptable efficiency in the influence spread and runtime than other algorithms.

**Keywords:** influence maximization; periphery node; shell-based ranking; seed node; social network

### Introduction

Social network analysis has many applications in diffusing ideas on social networks such as viral marketing, outbreak detection, epidemic spread, and word-of-mouth marketing, which is attracted by researchers in recent years.<sup>1–3</sup> The information spread process tries to expand information among nodes as far as possible.<sup>4</sup> For example, in word-of-mouth marketing, a company can have extensive advertising on its products by selecting a set of influential people on social networks.<sup>5,6</sup> One of the most important topics in information spreading is the influence maximization problem (IMP). The problem of selecting minimum influential nodes is known as the IM.<sup>6,7</sup> Notably, IM was first proposed by Domingos and Richardson, and then, Kempe et al. formulated this problem by examining two independent cascade (IC) and linear threshold (LT) models. It is notable that these models still are popular and used by researchers in many applications.<sup>8–14</sup>

The optimum information spreading depends on the selected influential nodes on the networks. It plays an im-

portant role in the formation of behavioral patterns and opinions on the social networks, which can be used to determine the initial influential nodes, the optimum spread of the opinions for advertising, and improve the formation of the opinion models.<sup>15</sup> IM refers to the selecting  $k$ -influential nodes that together have the highest effect on other nodes in total.<sup>16,17</sup> Theoretically and practically, the IMP provides new insights into the viral marketing,<sup>1,17–20</sup> outbreak detection,<sup>21</sup> rumor monitoring,<sup>22</sup> identification of the social network leaders, and so forth.

Diffusion models are one of the important problems in the field of network science that are used to map the IMP to information diffusion in the real world. IC and LT models are two widely used models in the IMP.<sup>1,4,5</sup> In the IC model, each edge has a probability of diffusion that each node independently activates the other node based on the probability of diffusion. In the LT model, each edge has a diffusion weight, and one node is activated if the total diffusion weight of the in-neighbors of the node is greater than the threshold. In IM, the

Department of Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran.

\*Address correspondence to: Asgarali Bouyer, Department of Computer Engineering, Azarbaijan Shahid Madani University, Tabriz 5971673831, Iran, E-mail: a.bouyer@azaruniv.ac.ir

function of the influence spread in the IC model and the LT is a NP-hard problem and it has submodularity and monotonicity properties.<sup>1,6</sup>

The IMP has recently been studied in many fields.<sup>23–28</sup> However, provided algorithms in recent years have several important problems. For example, most of the recently proposed algorithms do not consider the different impact of core nodes in the influence spread process because some core and high-degree nodes cannot provide optimal influence. In addition, these algorithms do not use a pruning approach at initial steps for the periphery and weak nodes to reduce computational time. To resolve these difficulties, this article proposes a new algorithm for solving the IMP under the IC model, which is more efficient in terms of runtime and influence spread than the other recently developed algorithms. This method is called the shell-based ranking and filtering method (SRFM) algorithm that is based on the hierarchical selection of the seed nodes and entails two general steps: the SRFM algorithm first divides the G graph into different sections, and graph segmentation is performed by the k-shell algorithm. After discovering all shells and core, the nodes are prioritized, and a pruning approach to remove periphery nodes is performed. Then, a set of candidate nodes is selected based on the assortativity and the range of influence (ROI) measures. In the second step, the seed nodes are selected based on the topological criteria from the candidate set. Experimental results on the synthetic and real-world networks show that the SRFM algorithm has better performance than the basic and other recently proposed algorithms in terms of the influence spread and runtime time.

The main contribution of this research is presented as (1) a new algorithm based on the hierarchical selection of the seed nodes is provided, which increases the algorithm efficiency in terms of the spread rate. (2) The influence spread calculations in the periphery nodes are optimally ignored in this algorithm, which reduces the computational overhead. (3) This algorithm examines the role of the core nodes in the influence spread by the influence power and the ROI measures. A small number of nodes can play an important role in diffusion, which are identified using the influence power and the ROI measures.

The structure of the rest of the article is presented hereunder. Related Studies section reviews some basic and new studies in the IMP. The Proposed Method section describes our proposed method. Experimental Results and Analysis section presents the experimental results to evaluate the SRFM algorithm. Finally, in Conclusion and Future Directions section some conclusions are reached.

## Related Studies

There are some survey articles related to the IMP.<sup>10,29,30</sup> This section reviews the algorithms presented in recent years. Maghami and Sukthankar proposed the hierarchical influence maximization (HIM) algorithm, which creates a local network for each node consisting of its neighbors and neighbors of neighbors.<sup>31</sup> In this algorithm, influential nodes are transferred to the next level of the hierarchy. Then, influential nodes select based on a convex optimization technique. This algorithm is not fast because it uses convex optimization. Kitsak et al. proposed the k-core algorithm to identify the influential nodes using their position in the network.<sup>32</sup> The k-core algorithm divides the network into different layers and shows that nodes in the same layers have the same influence spread. This algorithm is fast, but it does not have a suitable performance for selecting seed set in all networks.

Narayanam et al. proposed the SPIN algorithm based on game theory.<sup>33</sup> The Shapley value based influential nodes (SPIN) algorithm improves the runtime of the basic greedy algorithm,<sup>6</sup> but it does not provide an approximation guarantee. Therefore, the accuracy of the influence spread of the SPIN algorithm is lower than the greedy algorithm. Goyal et al. introduced a new model of credit distribution to find that the probability of selecting edges with different methods has a different effect on the selection of influential nodes.<sup>34</sup> Goyal also demonstrated that the credit distribution model for the IMP is an NP-hard problem.

Wang et al. proposed the maximum influence arborescence (MIA) scalable algorithm.<sup>35</sup> This algorithm was developed to avoid using unsuitable Monte Carlo simulation for online social networks due to its extremely low performance in large-scale networks. The MIA algorithm calculates the influence spread on the arborescence structure with an efficient recursion with suitable runtime. Since the MIA algorithm needs to store the data structure related to the local influence of each node, it likely has a high memory overhead. Therefore, Jung et al. proposed the influence ranking and influence estimation (IRIE) algorithm, which uses iterative computations to calculate the influence spread of the out-neighbors of each node, and during the computation steps, it does not require to store data structures. It has the low memory overload.<sup>36</sup>

Chen and colleagues studied the time criterion in the IMP.<sup>37</sup> They also revealed that the IMP has a submodularity property in the time-delayed IC model. Cheng et al. introduced the StaticGreedy algorithm to improve

the runtime of the NewGreedyIC algorithm.<sup>38</sup> The algorithm calculates the influence spread using the reachable nodes. Although this algorithm is fast, it does not provide an approximation guarantee. In addition, Morone et al. proposed the Collective Influence algorithm to choose the influential nodes with optimal percolation.<sup>39</sup> The algorithm locally calculates the influence spread in a circle of radius  $l$  (direct neighbors) for each node. The runtime of the algorithm is not suitable for the large-scale social networks.

Zhang et al. presented the VoteRank algorithm, which is based on the voting criterion.<sup>40</sup> In this algorithm, influential nodes are selected based on the voting ability of the neighbors of nodes. Although this algorithm is fast, it does not have acceptable accuracy for the influence spread. Therefore, Nguyen et al. introduced the ProbDeg algorithm that determines the effect of multihops on the IMP.<sup>41</sup> The algorithm is fast but does not ensure optimal approximation. Then, Zareie and Sheikhhahmadi proposed the hierarchical k-shell (HKS) algorithm, which calculates the spreading capability of nodes based on their locations and degrees as well as neighbors' information.<sup>42</sup> Also, this algorithm attempts to assign a hierarchical index for each node. The selection of seed nodes in this algorithm is not optimal because it ignores relations in high-density regions.

After that, Sun et al. developed the reversed local path (RLP) algorithm, which calculates the ability of the influence spread of the nodes using the local path.<sup>43</sup> The RLP algorithm can efficiently find influential nodes that can activate the localized target nodes. Localized target nodes are a set of nodes with high connections. Liu et al. presented the local index rank (LIR) algorithm.<sup>44</sup> LIR can detect the influential nodes using the local index and node degree criteria. The algorithm has a suitable runtime, but it does not have appropriate accuracy for the influence spread.

Li et al. proposed the Hierarchy based Influence Maximization (HBIM) algorithm, which divides the information diffusion into levels, and finds influential nodes in each level based on random walk to calculate the belonging coefficient.<sup>45</sup> The greedy algorithm is better than the HBIM algorithm in terms of influence spread. Xin et al. proposed the heterogeneity-oriented (HO) centrality to find the influential nodes.<sup>46</sup> The algorithm uses two criteria of the spread rank and activity rank in this centrality to find the influential nodes for immunization. Talukder et al. introduced the KIRM algorithm under the LT model.<sup>47</sup> In the KIRM algorithm, the nodes activation process ends with the concept of influence decay.

Qiu et al. proposed the Partition-Heuristic-Greedy (PHG) algorithm that uses the community detection method and greedy algorithm.<sup>48</sup> This algorithm achieves excellent stability in influence spread. Beni and Bouyer presented the top-k influential nodes selection based on community detection and scoring criteria in social networks (TI-SC) algorithm to improve the spread of the DegreeDiscount algorithm.<sup>14</sup> The algorithm selects the seed nodes using a community detection method and a scoring measure. The main advantage of the algorithm is that it uses a scoring system and thus selection of the seed nodes is closer to the real world. However, it takes a long time to run on the large-scale graphs.

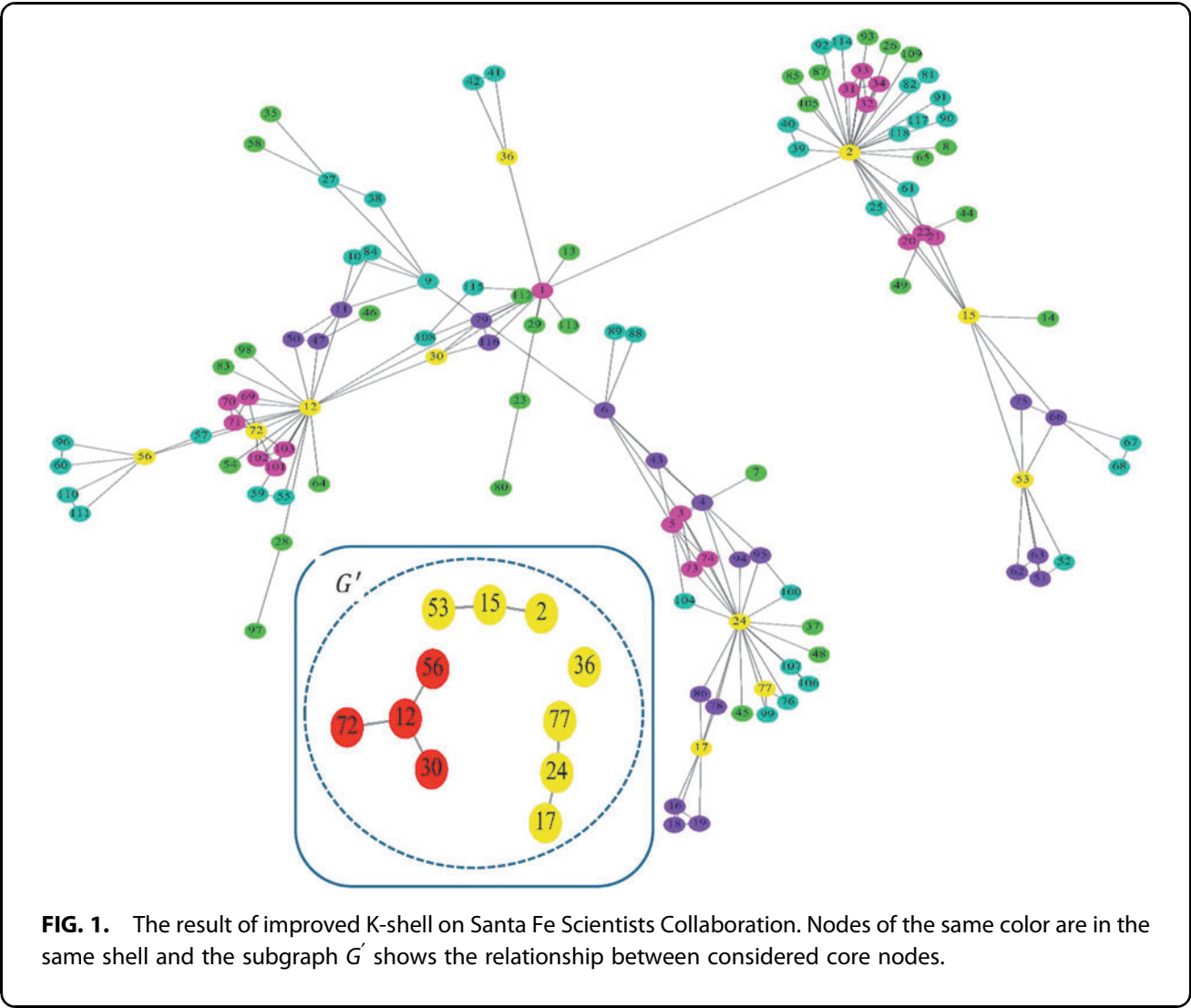
Ding et al. provided the D-greedy algorithm under the realistic independent cascade (RIC) model for the IMP.<sup>49</sup> The main advantage of the algorithm is that the IMP under the RIC model is suitable for analyzing the real-world problems. Aghaee and Kianian proposed Group of Influential Nodes (GIN) algorithm that creates different groups of graph nodes and by using the Expected Diffusion Value selects influential nodes.<sup>50</sup> In some social networks, the greedy algorithm has a better influence spread than the GIN algorithm. The GIN algorithm has a good efficiency.

### The Proposed Method

In the IMP, the position of a node on the core and other shells is important on the selection of seed nodes in social networks. In the proposed SRFM algorithm, the seed nodes are, therefore, selected from the core and other important nodes based on their position. The SRFM algorithm has two main steps: (1) hierarchical selection of the initial seed nodes and (2) selection of the final seed nodes.

#### Hierarchical selection of the initial seed nodes

In this step of the algorithm, the graph  $G=(V, E)$  is initially divided into different parts. The graph shells are specified by the modified k-shell algorithm. In the modified k-shell algorithm, nodes with degree 1 are initially removed from the graph  $G$  and placed at a depth of 1 of shell 1. By removing the nodes with degree 1, another node with degree 1 is created in the graph, which is placed in depth 2 of shell 1, and this process continues until the new degree 1 node is not created in the graph. Then, nodes with degree 2 are removed from the graph  $G$  and placed at the depth 1 of shell 2. By removing the nodes with degree 2, other nodes with degree 2 are possibly created in the graph, which are placed at a depth of 2 of shell 2, and this process continues until there is no node with degree 2 in the graph. The overall process of the k-shell algorithm continues until all



nodes are placed in related shells. For example, in Figure 1, nodes with similar colors are placed in the equal shells.

According to Table 1, all the nodes in this Network are identified by the shell depth. After detecting all the shells, based on the remaining edges between the core nodes, the graph  $G' = (V', E')$  is established. As shown in Figure 1, the graph  $G'$  is obtained for sample Network so that the red nodes are the nodes in the largest component. Therefore, for each graph  $G'$ , with  $V' = \{v'_1, v'_2, v'_3, \dots, v'_n\}$  and  $E' = \{e'_1, e'_2, e'_3, \dots, e'_m\}$ , if  $v'_i \in V'$  the influence power of node  $v'_i$  ( $PI_{v'_i}$ ) is calculated according to the following equation:

$$PI_{v'_i} = \frac{co_{v'_i} + (d_{v'_i, G'})^2}{\Delta(A)_{v'_i}}, \quad (1)$$

where  $co_{v'_i}$  is the number of edges between  $v'_i$  with other nodes in the other shells,  $d_{v'_i, G'}$  is the degree of node

**Table 1. Separation of nodes based on the depth and shell in the Santa Fe Scientists Collaboration Network**

Shell	Color	Depth	Nodes
1	Green	1	{13, 29, 112, 113, 8, 26, 65, 85, 87, 93, 105, 109, 7, 54, 64, 83, 98, 14, 44, 49, 80, 37, 45, 48, 35, 58, 97, 46}
		2	{23, 28}
2	Turquoise	1	{115, 25, 39, 40, 61, 81, 82, 90, 91, 92, 114, 117, 118, 104, 88, 89, 10, 27, 38, 84, 55, 57, 59, 76, 99, 100, 106, 107, 41, 42, 52, 60, 96, 110, 111, 67, 68}
		2	{108, 9}
3	Purple	1	{79, 116, 43, 94, 95, 11, 47, 50, 66, 75, 16, 18, 19, 78, 86, 51, 62, 63}
		2	{6}
4	Pink	3	{4}
		1	{1, 20, 21, 22, 31, 32, 33, 34, 3, 5, 73, 74, 69, 70, 71, 101, 102, 103}

$v_i \in V'$  in the graph  $G'$ , and  $\Delta(A)_{v_i}$  represents the assortativity criterion changes after removing the node  $v_i \in V'$  in the graph  $G'$ . If node  $v_i \in V'$  has a high influence power by deleting the node  $v_i$ , the  $\Delta(A)_{v_i}$  criterion is small. In other words, it has less changes by deleting the node  $v_i$ . However, if the node  $v_i$  has lower influence power by deleting the node  $v_i \in V'$ , the criterion  $\Delta(A)_{v_i}$  experiences many changes and  $\Delta(A)_{v_i}$  increases in this case. After calculating  $PI_{v_i}$ , the candidate set from the core nodes called must be selected that is called CC set.

**Definition 1.** The ROI: For the graph  $G'$ , the criterion  $f$  to determine the influential power range is expressed as follows:

$$f = \frac{2k + b_{lc}}{\log k}, \quad (2)$$

where  $k$  is the size of a set of the seed nodes and  $b_{lc}$  is the number of the nodes in the largest component connected to the graph  $G'$ . If  $f > b_{lc}$ , then  $ROI = b_{lc}$ , also if  $f < b_{lc}$ , then  $ROI = f$ .

$$ROI = \min(b_{lc}, f). \quad (3)$$

**Definition 2.** The CC set:  $A_{G'}$  is the assortativity coefficient for the graph  $G$ . The  $v^*$  set is known as the CC set with the ROI size where

$$PI(v^*) = \begin{cases} \operatorname{argmax}_{v' \subseteq V \setminus |v'| < ROI} PI(v'), & A_{G'} \geq 0 \\ \operatorname{argmin}_{v' \subseteq V \setminus |v'| < ROI} PI(v'), & A_{G'} < 0 \end{cases} \quad (4)$$

---

Algorithm 1: Discovering set  $cc$  of the network ( $G$ )

---

**Input:** Network  $G (V, E)$

**Output:** Set  $cc$

```

1: initialize  $cc \rightarrow \emptyset$ ;
2: Find the graph  $G' = (V', E')$  by the k-shell algorithm.
3: for each  $v_i$  in  $b_c$  do
4:    $PI_{v_i} = \frac{c_{v_i} + (d_{v_i, G'})^2}{\Delta(A)_{v_i}}$ 
5: end for
//discovering  $cc$  set in network  $G'$  ( $v^*$  set)
6: if  $A_{G'} > 0$  then
7:    $PI(v^*) = \operatorname{argmax}_{v' \subseteq V \setminus |v'| < ROI} PI(v')$  // largest  $PI_{v_i}$  value and with the size of ROI
8: end if
9: if  $A_{G'} < 0$  then
10:   $PI(v^*) = \operatorname{argmin}_{v' \subseteq V \setminus |v'| < ROI} PI(v')$  // smallest  $PI_{v_i}$  value and with the size of ROI
11: end if
12: return Set  $cc$ 
```

---

Therefore, Algorithm 1 shows the creating of the CC set. First, the graph  $G' = (V', E')$  is created in line 2 based on the improved  $k$ -shell algorithm. In lines 3–5 for each node  $v_i$  in the largest component ( $b_c$ ) of the graph  $G'$ , the criterion  $PI_{v_i}$  is calculated. In lines 6–8, if  $A_{G'} > 0$ , then the  $v^*$  set is selected with the size of ROI from  $v_i$  nodes with the largest  $PI_{v_i}$  value and is known as the CC set. In lines 9–11, if  $A_{G'} < 0$ , then the  $v^*$  set with the smallest  $PI_{v_i}$  value and with the size of ROI is selected as the CC set. After producing the CC set, the PC set is generated for located nodes in other shells. Before producing this set, the nodes in  $1 \leq k_s \leq limit$  are completely ignored, and the limit is defined as follows:

$$limit = \frac{n_s}{A_d} + 1, \quad (5)$$

where  $n_s$  is the number of the graph shells and  $A_d$  is defined according to equation (6).

$$A_d = \frac{\sum_{i=1}^{n_s} depth_i}{n_s}, \quad (6)$$

where  $depth_i$  is the number of the depth of the shell  $i$ . After filtering some unimportant shells, the nodes in the remaining shells are sensitively examined for each depth to add suitable nodes to the PC set. At each depth of a shell, the nodes with a lower clustering coefficient may have more influence spread than the other existing nodes of the same depth. If a node inside the shell satisfies equation (7), then it is added to the PC set.

$$c_{v_i} < \frac{\sum_{i=1}^{n_d} c_i}{n_d}, \quad (7)$$

where  $c_{v_i}$  is the clustering coefficient of a node  $v_i$  and  $n_d$  is the number of nodes in a desired shell. If the value  $c_{v_i}$  for node  $v_i$  at the depth of a shell is less than  $\frac{\sum_{i=1}^{n_d} c_i}{n_d}$ , the node  $v_i$  is added to the PC set. Moreover, if the CC set is  $CC = \{v_{c,1}, v_{c,2}, v_{c,3}, \dots, v_{c,l}\}$  and  $\Gamma(v_{c,i})$  is a set of the neighbors for a node  $v_{c,i}$  in the CC sets, then, if the node  $v_i \in \Gamma(v_{c,i})$  does not satisfy the condition in equation (7), but it satisfies the condition  $d_{v_i, G} \geq \frac{\sum_{i=1}^{n'} d_{v_i, G'}}{n'}$ , in this case, the node  $v_i$  is added to the PC set, because the node  $v_i$  can play an important social role between the cores and other nodes and helps to diffuse information on the network, effectively. In addition, it should be noted that the nodes in the

shells  $k_s = n_s - i$  where  $i=1, 2$ , and  $3$  have close relationships with the core nodes, all nodes in these shells are also added to the PC set.

Algorithm 2: Discovering set PC of network (G)

---

**Input:** Network G (V, E) and set CC  
**Output:** Set PC  
1: initialize  $PC \rightarrow \emptyset$ ;  
2: Find the shells of the graph  $G=(V, E)$  by the k-shell algorithm.  
3: **for each**  $k_s$  in shells **do**  
4:   **if**  $1 \leq k_s \leq limit$  **then**  
5:     pass  
6:   **end if**  
7:   **if**  $k_s = n_s - i$  **then** //  $i=1, 2, 3$   
8:     all nodes in  $k_s$  add to the set PC  
9:   **end if**  
10:   **for each**  $d_i$  in shell **do**  
11:     **for each**  $v_i$  in depth **do**  
12:       **if**  $c_{v_i} < \frac{\sum_{i=1}^{n_d} c_i}{n_d}$  **then**  
13:          $PC \leftarrow v_i$   
14:       **end if**  
15:       **if**  $c_{v_i} > \frac{\sum_{i=1}^{n_d} c_i}{n_d}$  &&  $v_i \in \Gamma(v_{c,i})$  &&  $d_{v_i,G} \geq \frac{\sum_{i=1}^{n'} d_{v_i,G'}}{n'}$   
**then**  
16:          $PC \leftarrow v_i$   
17:       **end if**  
18:     **end for**  
19:   **end for**  
20: **end for**  
21: **return** Set PC

---

Hence, Algorithm 2 is proposed to produce the PC set by the SRFM algorithm. In lines 4–6, the nodes inside the  $1 \leq k_s \leq limit$  shells are first completely ignored. In lines 7–9, all nodes in the  $k_s = n_s - i$  shells where  $i=1, 2$ , and  $3$  are added to the PC set. In lines 10–14, if the  $c_{v_i}$  value for the node  $v_i$  at the shell depth is less than  $\frac{\sum_{i=1}^{n_d} c_i}{n_d}$ , the node  $v_i$  in line 13 is added to the set PC. In lines 15–17, if the value  $c_{v_i}$  for node  $v_i$  at the shell depth is higher than  $\frac{\sum_{i=1}^{n_d} c_i}{n_d}$  and  $v_i \in \Gamma(v_{c,i})$  and it satisfies the condition  $d_{v_i,G} \geq \frac{\sum_{i=1}^{n'} d_{v_i,G'}}{n'}$ , the node  $v_i$  is added to the PC set in line 16.

#### Selecting the final seed nodes

After creating the CC and PC sets, the CG set must be generated to select the seed nodes. Accordingly, the CG set is produced according to the following equation:

$$CG = CC \cup PC. \quad (8)$$

At next, for each node  $v_i$  in the CG set, equation (9) is calculated. Then the k node with the highest  $IN_{v_i}$  value is selected as the final seed set.

$$IN_{v_i} = \frac{\Gamma(v_i)^2}{d_{max} \left( \sum_{i=1} \left( \frac{|\Gamma(v_i)|}{|\Gamma(v_i) \cup \Gamma(u_i)| - |\Gamma(v_i) \cap \Gamma(u_i)|} \right)^2 \right)}, \quad (9)$$

where  $d_{max}$  is the maximum degree in the graph  $G'$  and  $\Gamma(v_i)$  is the neighbors of the node  $v_i$ .  $v_i \rightarrow u_i$  is the edge between the nodes  $v_i$  and  $u_i$ .

Therefore, Algorithm 3 is introduced for the second step of the SRFM algorithm. The CG set is generated in line 2. In lines 3–5, the criterion  $IN_{v_i}$  is calculated for each node  $v_i$  in the CG set. Moreover, the nodes are arranged in descending order and in line 7, and the k node with the highest  $IN_{v_i}$  value is added to the final seed set.

Algorithm 3: Selecting the seed nodes of network (G)

---

**Input:** Network G (V, E), set CC and set PC  
**Output:** Set Seed nodes  
1: initialize seeds nodes  $\rightarrow \emptyset$ ;  
2:  $CG = CC \cup PC$   
3: **for each**  $v_i$  in CG **do**  
4:    $IN_{v_i} = \frac{\Gamma(v_i)^2}{d_{max} \left( \sum_{i=1} \left( \frac{|\Gamma(v_i)|}{|\Gamma(v_i) \cup \Gamma(u_i)| - |\Gamma(v_i) \cap \Gamma(u_i)|} \right)^2 \right)}$   
5: **end for**  
6: Rank the values of  $IN_{v_i}$  in descending order  
7: select the number of k nodes with  $IN_{v_i}$  largest and add to set seed node  
8: **return** Set seed nodes

---

#### Time complexity analysis

The time complexity analysis in the SRFM algorithm consists of two steps. In the first step, time complexity refers to the hierarchical selection of the initial seed nodes. The time complexity for this step is  $O(m)$ , where m is the number of the edges in the graph. In the second step, time complexity represents the selection of the final seed set that is  $O(n')$ , where  $n'$  is the number of nodes in the CG set and  $n' < n$  and n is the number of the nodes in the network. Thus, the total time complexity of the SRFM algorithm is  $O(m + n')$ .

#### Experimental Results and Analysis

In this section, we use 10 networks (2 synthetic networks and 8 real networks) with different sizes and categories to evaluate the effectiveness and efficiency of SRFM compared with the other 7 well-known algorithms. Moreover, the fundamental features for these networks are given in Table 2.

**Table 2. Summary of eight real-world networks and two synthetic networks**

Networks	Nodes	Edges	Maximum degree	Minimum degree	90% Effective diameter	Diameter
Route views	6474	13,895	1459	1	4.4	9
PGP	10,680	24,316	205	1	10.1	24
Sister cities	14,274	20,573	99	1	10.2	25
As-22july06	22,693	48,436	2390	1	4.6	9
COND-MAT-2003	23,133	93,497	560	1	6.5	14
CAIDA	26,475	53,381	2628	1	4.6	17
COND-MAT-2005	39,577	175,691	278	1	6.7	18
Douban	154,908	327,162	287	1	5.6	9
M-FO115	10,000	23,142	155	1	11.8	26
M-FO120	10,000	29,566	229	1	11.6	25

CAIDA, Center for Applied Internet Data Analysis; PGP, Pretty Good Privacy.

### Real-world networks

Here are the descriptions of the eight real-world data sets. All data sets are undirected. Data sets of these eight networks can be downloaded from the Konect website.\*

- Route views: This is the network of autonomous systems of the Internet connected with each other.<sup>51</sup>
- PGP: This network is the interaction network of users of the Pretty Good Privacy (PGP) algorithm.<sup>52</sup>
- Sister cities: The Sister cities network is an undirected network of cities of the world connected by “sister city” relationships, as extracted from WikiData.<sup>51</sup>
- As-22july06: The As-22july06 is a network of the structure of the Internet at the level of autonomous systems.<sup>51</sup>
- COND-MAT-2003: The COND-MA (1993–2003) collaboration network is from the e-print arXiv and covers scientific collaborations between author’s articles submitted to Condense Matter that the data cover articles in the period from January 1993 to April 2003.<sup>53</sup>
- Center for Applied Internet Data Analysis: The CAIDA is the network of autonomous systems of the Internet connected with each other from the CAIDA project.<sup>51</sup>
- COND-MAT-2005: The COND-MA (1995–2005) collaboration network is from the e-print arXiv and covers scientific collaborations between author’s articles submitted to Condense Matter that the data covers articles in the period from 1995-01-01 to April 2005-03-31.<sup>51</sup>
- Douban: The Douban network is a Chinese social networking service of a Chinese online recommendation site.<sup>54</sup>

### Synthetic networks

Here are the descriptions of the two synthetic networks, and the forest fire model is used to make synthetic networks. The power-low distribution property is important in the forest fire model. Synthetic networks are as follows:

- M-Fo115: This network created from the forest fire model with connecting probability  $p=0.115$ .<sup>55</sup>
- M-Fo120: This network created from the forest fire model with connecting probability  $p=0.120$ .<sup>55</sup>

### Basic compared algorithms

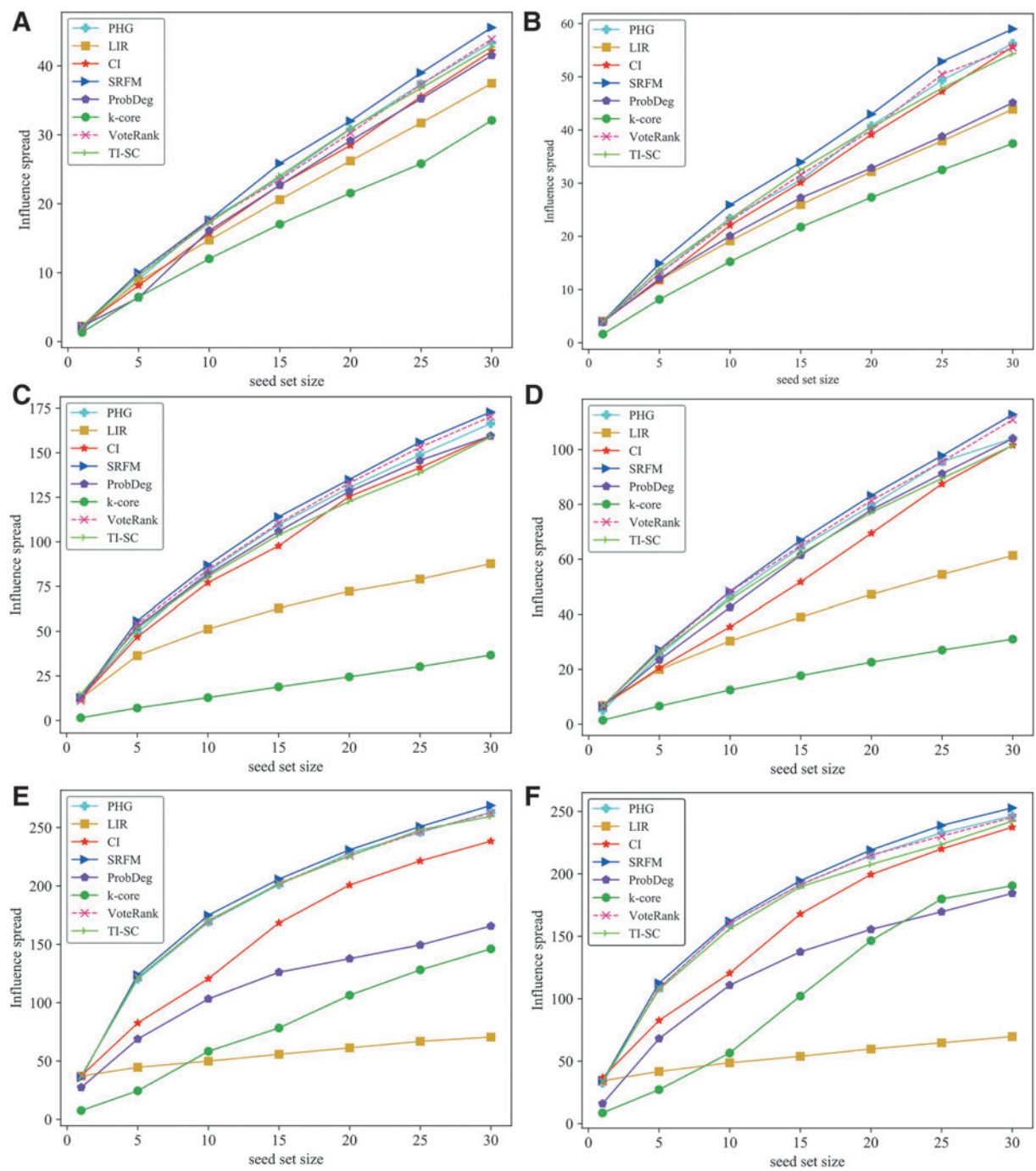
The SRFM algorithm has been compared with seven baseline algorithms (i.e., TI-SC, PHG, LIR, ProbDeg, VoteRank, CI, and k-core). The list of baseline algorithms is described hereunder.

- TI-SC: This algorithm is a community-based approach. It selects the seed nodes based on the special scoring system near to real-world ranking.<sup>14</sup>
- PHG: This algorithm is based on the community detection method and greedy algorithm.<sup>48</sup>
- LIR: Influential nodes are selected in the LIR algorithm using the criteria of local index and node degree.<sup>44</sup>
- ProbDeg: The ProbDeg determines the effect of multihops on the IMP.<sup>41</sup>
- VoteRank: In the VoteRank algorithm, influential nodes are selected based on the voting ability of the neighbors of one node.<sup>40</sup>
- CI: The CI algorithm locally calculates the influence spread in a circle of radius  $l$  for each node.<sup>39</sup>
- K-core: The k-core algorithm divides the network into different layers and shows that nodes in the same layers have the same influence spread.<sup>32</sup>

### Experiment setup

In this article, the SRFM algorithm is programmed in python language and is ran on a computer with

\*<http://konect.cc/>



**FIG. 2.** Influence spread comparison in various data sets under independent cascade model. **(A)** Sister cities; **(B)** PGP; **(C)** COND-MAT (1993–2003); **(D)** COND-MAT (1995–2005); **(E)** CAIDA; **(F)** As-22july06; **(G)** Douban; **(H)** Route views; **(I)** M-Fo115; **(J)** M-Fo120. CAIDA, Center for Applied Internet Data Analysis; PGP, Pretty Good Privacy.



2.5GHz Intel Core i5 CPU-3230M and 16GB memory. Our proposed algorithm supports instances of the IC model. In this model, influence probability of each edge is  $p_{uv}=0.01$ . To obtain the influence spread of these algorithms, we simulate the IC model on the networks 1000 times and take the average of the influence spread.

### Experimental results

**Quality.** Quality in the IMP is the amount of the influence spread of seed nodes. The high influence spread of the algorithm indicates the high quality of the algorithm. We select seven distinct sets of seed users of size 1, 5, 10, 15, 20, 25, and 30 for 10 data sets. Figure 2 shows the amount of the influence spread of SRFM with state-of-the-art algorithms under IC models, where the x-axis represents the number of seed

nodes. In contrast, the y-axis represents the overall influence spread. The results in real-world data sets and synthetic networks represent that the SRFM algorithm completely outperforms other state-of-the-art algorithms in terms of influence spread. Compared with the mentioned algorithms, the selected seed set by the SRFM algorithm has the largest influence spread.

Figure 2A shows the experimental results of the Sister cities data set which we can observe significant gaps of influence spread value between the SRFM algorithm and other algorithms. The k-core and LIR algorithm show the worst performance in Figure 2A. Figure 2B shows the influence spread of the SRFM algorithm in the PGP data set. In Figure 2B, SRFM is better than PHG, VoteRank, and TI-SC on the PGP network. Figure 2C shows that the influence spread of the SRFM algorithm in the COND-MAT (1993–2003) is the best. The k-core algorithm shows the worst performance in the

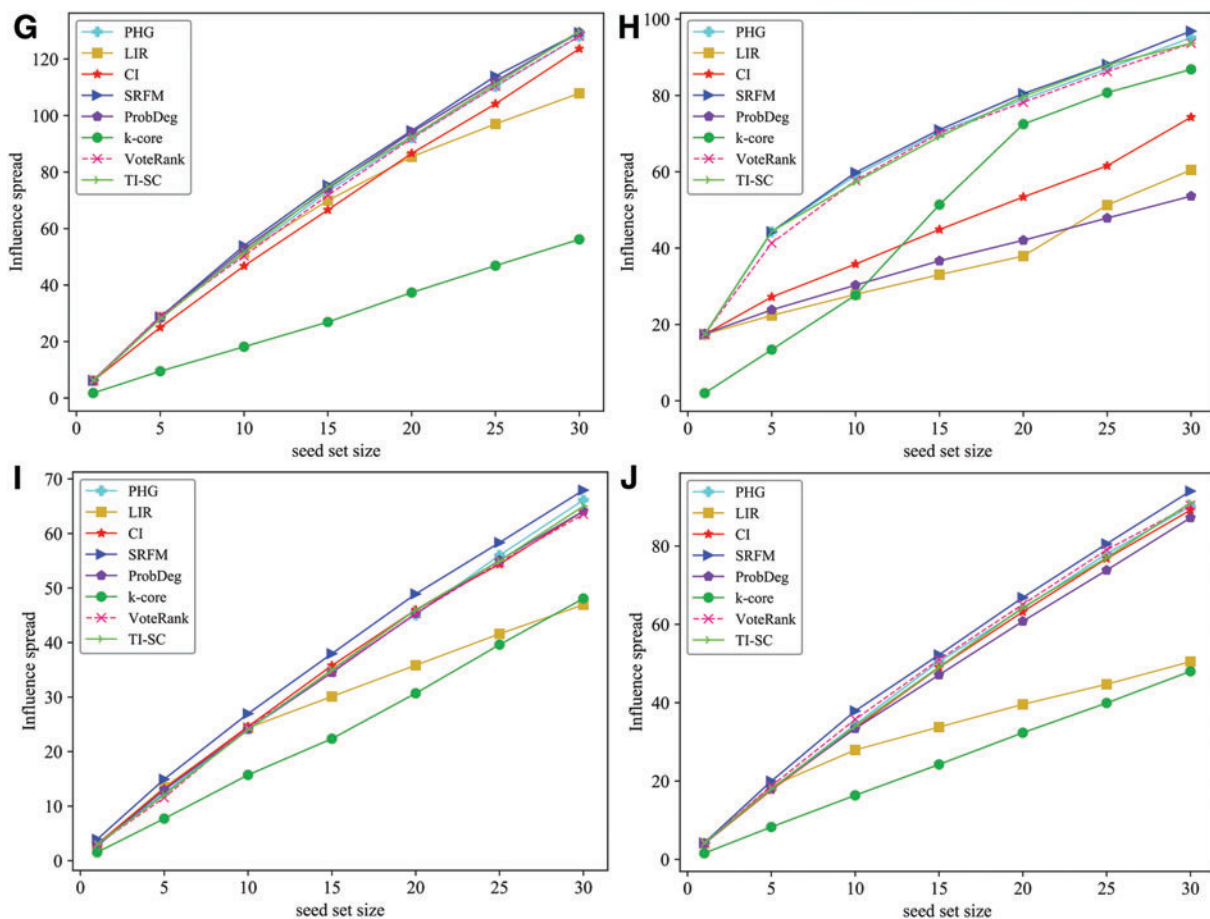
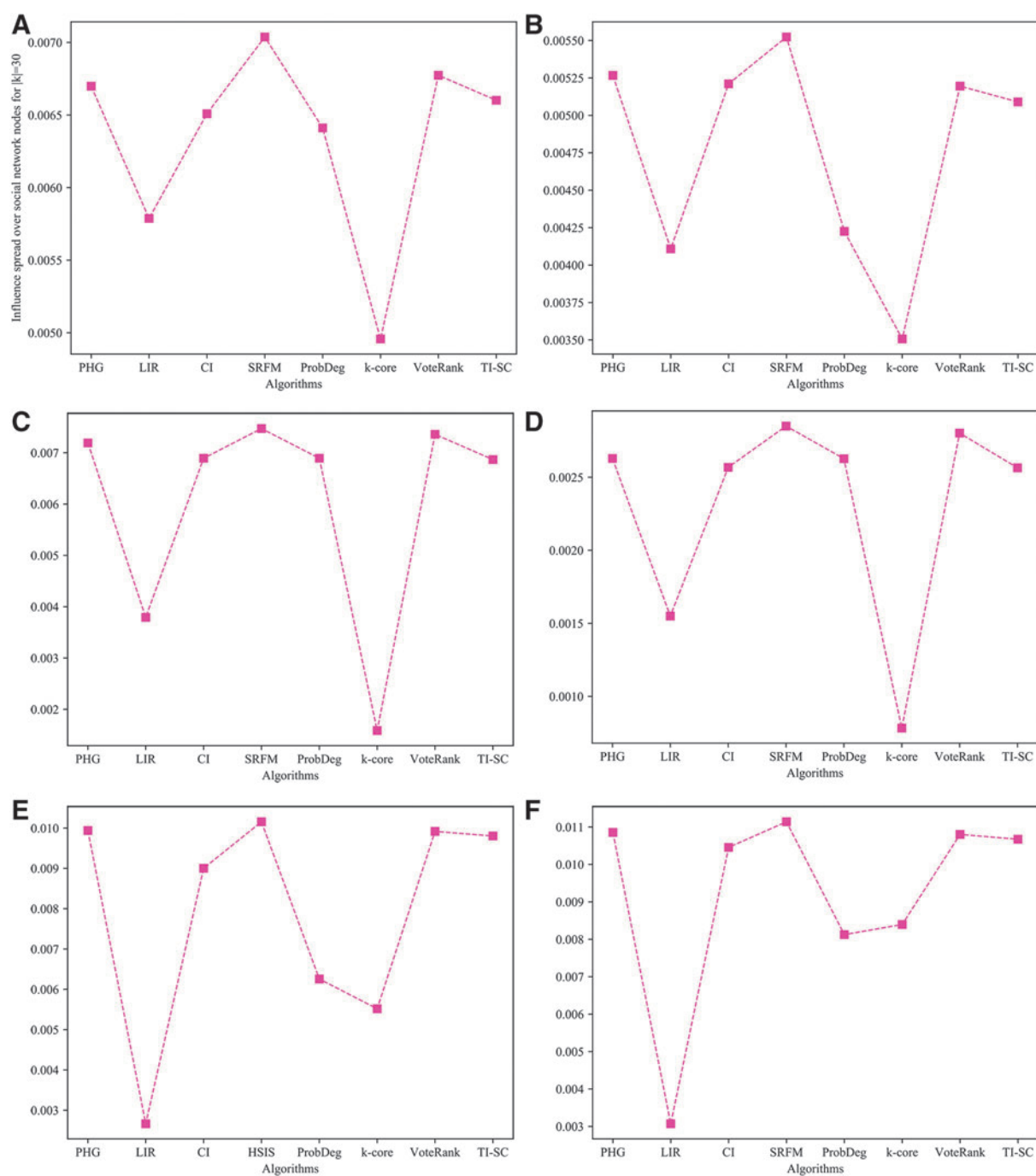


Fig. 2. (Continued).



**FIG. 3.** The influence spread over social network nodes comparison in various data sets under independent cascade model for  $|k|=30$ , (A) Sister cities; (B) PGP; (C) COND-MAT (1993–2003); (D) COND-MAT (1995–2005); (E) CAIDA; (F) As-22july06; (G) Douban; (H) Route views; (I) M-Fo115; (J) M-Fo120.

COND-MAT-2003 data set. Figure 2D shows the influence spread of the SRFM algorithm in the CONDMAT-2005 data set which we see that the influence spread values of TI-SC, PHG, and VoteRank algorithms are lower than the SRFM algorithm. Figure 2E shows the influence spread of the SRFM algorithm in the CAIDA data set which the SRFM algorithm outperforms other algorithms. Likewise, PHG, TI-SC, and VoteRank algorithms have the same influence spread value shown in Figure 2E. Figure 2F shows the influence spread of the SRFM algorithm in the As-22july06 data set. The SRFM algorithm is superior to the VoteRank, PHG, and TI-SC algorithms shown in Figure 2F. Also, the k-core algorithm shows the worst performance in Figure 2F. Figure 2G shows that the influence spread of the SRFM algorithm in the Douban data set. The influence spread values of the PHG, TI-SC, and VoteRank algorithms are the same as the SRFM algorithm in this data set. Finally,

Figure 2H shows the influence spread of the SRFM algorithm in the Route views data set. When  $k$  is 1, 10, and 15, the influence spread values of the PHG and VoteRank algorithms are the same as the SRFM algorithm. In Figure 2G and H, the influence spread of the compared algorithms is rather close when  $k$  is small values. We can observe significant gaps of influence spread value between the SRFM algorithm and other algorithms shown in Figure 2I. Figure 3J shows the influence spread of the SRFM algorithm in the M-FO120 data set. As can be seen in Table 3, SRFM is the best algorithm.

The influence spread is calculated based on the total data nodes for  $k=30$  in Figure 3 on real-world networks and synthetic networks. As shown in Figure 3, the influence spread of SRFM is better than other algorithms. We also analyze the average influence spread in  $k=1$ ,  $k=5$ ,  $k=10$ ,  $k=15$ ,  $k=20$ , and  $k=30$  on real-world networks and synthetic networks in

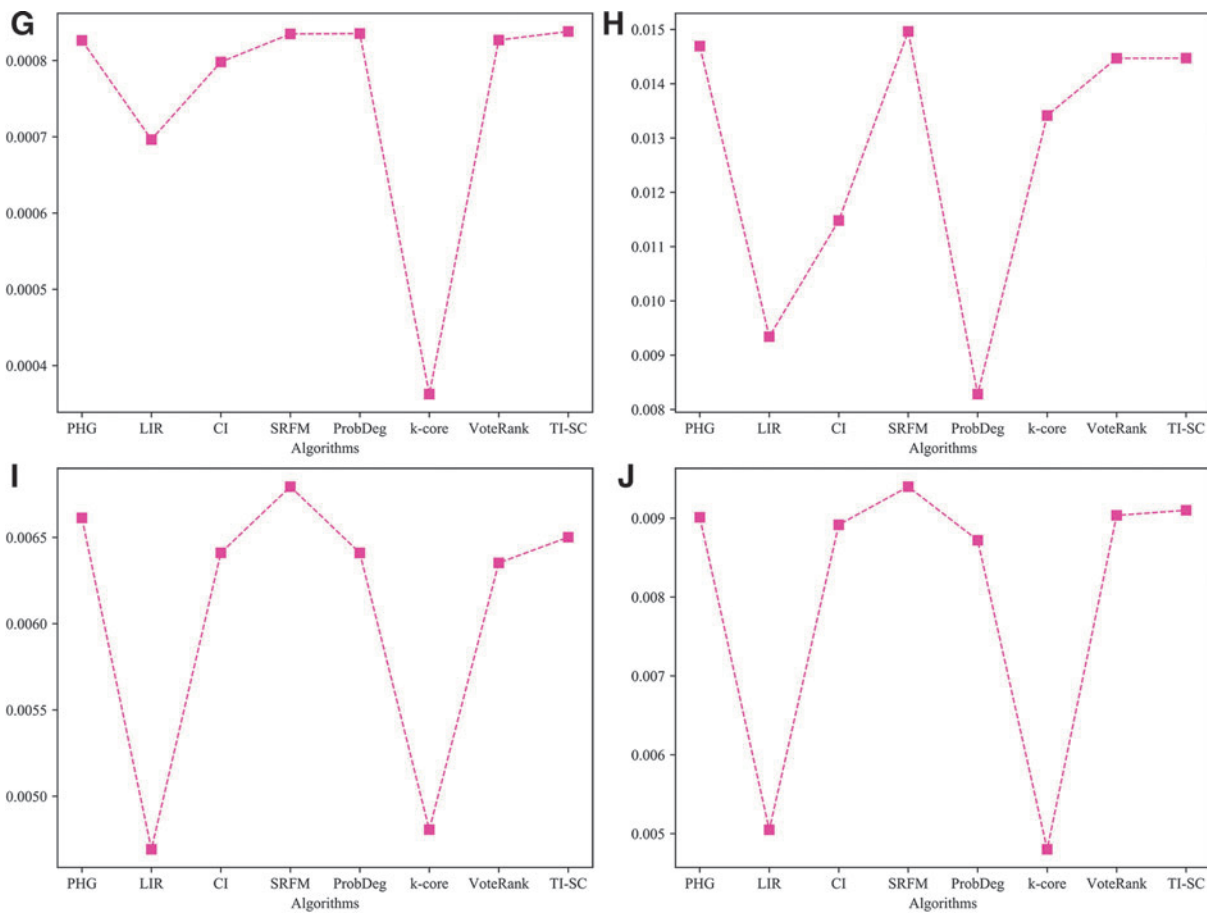


Fig. 3. (Continued).

**Table 3. The average number of influence spread in  $k=1$ ,  $k=5$ ,  $k=10$ ,  $k=15$ ,  $k=20$ , and  $k=30$  on eight real-world data sets and two synthetic networks**

<i>Algorithms</i> <i>Data sets</i>	<i>PHG</i>	<i>LIR</i>	<i>CI</i>	<i>SRFM</i>	<i>ProbDeg</i>	<i>k-Core</i>	<i>VoteRank</i>	<i>TI-SC</i>
Route views	15.07	8.34	10.48	<b>15.27</b>	8.39	11.15	14.82	14.84
PGP	7.24	5.83	7.00	<b>7.86</b>	6.00	4.80	7.25	7.21
Sister cities	5.45	4.72	5.16	<b>5.74</b>	5.10	3.87	5.47	5.44
As-22july06	39.54	12.43	35.48	<b>40.46</b>	28.06	23.71	39.46	38.69
COND-MAT (1993–2003)	23.34	13.39	22.02	<b>24.33</b>	22.89	4.38	23.87	22.37
CAIDA	42.15	12.87	35.65	<b>43.15</b>	25.95	18.31	42.19	42.12
COND-MAT (1995–2005)	14.00	8.63	12.44	<b>14.75</b>	13.57	3.96	14.48	13.62
Douban	16.30	14.88	15.29	<b>16.93</b>	16.56	6.55	16.26	16.47
M-Fo115	8.06	6.50	8.03	<b>8.60</b>	7.96	5.50	7.90	8.00
M-Fo120	11.30	7.30	11.10	<b>11.80</b>	10.80	5.60	11.40	11.20

The highest value of each row is bolded.

CI, Collective Influence; LIR, local index rank; PHG, Partition-Heuristic-Greedy; SRFM, Shell-based Ranking and Filtering Method; TI-SC, Top-k Influential nodes based on Scoring criteria and Community detection.

Table 3. As given in Table 3, we can easily find the SRFM algorithm is better than others.

**Computational time.** In this section, the running time of the SRFM algorithm is compared with that of the baseline algorithms. Table 4 gives the running time of different algorithms on the eight real-world networks and two synthetic networks. From the results, it is obvious that the running times of SRFM and k-core are lower than other compared algorithms. From the eight real-world data sets, SRFM is the best in Route views, Sister cities, and Douban data sets and k-core is the best in PGP, COND-MAT-2003, COND-MAT-2005, CAIDA, M-Fo115, and M-Fo120 data sets. However, as shown in Figures 2 and 3, k-core has the lowest quality in comparison with other algorithms, and it cannot provide any performance guarantee in terms of influence spread. Therefore, if we do not consider the k-core algorithm, our SRFM method in all data sets has the best computational time and performance compared with the rest. The SRFM algorithm is more time efficient than PHG, TI-SC, VoteRank, CI, and LIR algorithms.

In synthetic networks, k-core is the best and SRFM in the second best in comparison with other methods. Nevertheless, the SRFM in both M-Fo115 and M-Fo120 synthetic networks has the best quality, whereas k-core algorithm has the lowest quality in these data sets.

### Conclusion and Future Directions

In this article, we have presented a new algorithm for IMP, called SRFM. SRFM overcomes the low accuracy of the VoteRank, PHG, and TI-SC algorithms and the low efficiency of the CI and PHG algorithms based on the hierarchical selection of the seed nodes. The proposed algorithm examines the role of the core nodes and nodes within other shells in the influence spread and selection of the seed nodes. The influence spread calculations in the periphery nodes are optimally ignored in this algorithm, which reduces the computational overhead. Thus, the SRFM algorithm considers two main steps to improve the effectiveness of selected seed nodes: (1). Hierarchical selection of the initial seed nodes with aim to filter unimportant nodes and to

**Table 4. Execution times on real-world networks and synthetic networks (all times are in seconds)**

<i>Algorithms</i> <i>Data sets</i>	<i>PHG</i>	<i>LIR</i>	<i>CI</i>	<i>SRFM</i>	<i>ProbDeg</i>	<i>k-Core</i>	<i>VoteRank</i>	<i>TI-SC</i>
Route views	8.656E+05	2.608E+05	8.351E+07	<b>2.232E+05</b>	2.454E+05	6.211E+05	8.656E+05	2.707E+06
PGP	3.451E+08	1.780E+05	2.810E+07	8.898E+04	8.919E+04	<b>6.386E+04</b>	2.975E+05	1.262E+07
Sister cities	4.975E+05	1.411E+05	2.574E+05	<b>5.348E+04</b>	1.798E+05	6.116E+04	2.479E+05	9.443E+05
As-22july06	9.505E+08	5.229E+05	1.123E+09	<b>1.350E+06</b>	2.232E+06	1.370E+06	2.107E+06	9.431E+06
COND-MAT-2003	1.037E+08	1.576E+05	7.744E+07	4.208E+04	4.560E+05	<b>3.211E+04</b>	5.380E+05	1.311E+08
CAIDA	1.296E+09	4.927E+05	1.210E+09	1.753E+06	9.109E+05	<b>7.726E+05</b>	2.111E+06	5.672E+07
COND-MAT-2005	9.166E+09	7.571E+05	3.498E+08	8.664E+04	1.538E+06	<b>6.703E+04</b>	1.630E+06	7.561E+07
Douban	6.913E+08	8.556E+05	6.048E+05	<b>5.383E+05</b>	9.365E+05	1.076E+06	1.055E+06	4.390E+06
M-Fo115	8.230E+07	1.313E+05	8.760E+06	1.103E+05	2.657E+05	1.030E+05	2.288E+05	1.509E+05
M-Fo120	2.359E+08	2.585E+05	1.869E+07	2.382E+05	5.161E+05	<b>1.663E+05</b>	5.621E+05	2.746E+05

The lowest value of each row is bolded.

reach a fast approach in detecting the CC and PC sets and (2). the final seed set selection from the CC and PC sets. Since the number of nodes in these sets is insignificant, they are quickly processed and the final seed set is selected. The experimental results on eight real-world networks and two synthetic networks proved that the SRFM algorithm outperforms other algorithms in terms of influence spread (quality). Correspondingly, the SRFM algorithm is the fastest algorithm in 50% of data sets and is the second best in other 50% data sets. Consequently, the SRFM algorithm is a trade-off between time and efficiency. As a future study, the SRFM algorithm can completely be extended to make a parallel processing in finding the CC and PC sets and seed node selection step.

### Author Disclosure Statement

No competing financial interests exist.

### Funding Information

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

### References

- Chen W, Wang Y, Yang S. Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France: ACM, 2009. pp. 199–208.
- Liqing Q, Chunmei G, Shuang Z, et al. TSIM: A two-stage selection algorithm for influence maximization in social networks. *IEEE Access*. 2020;8: 12084–12095.
- Samadi N, Bouyer A. Identifying influential spreaders based on edge ratio and neighborhood diversity measures in complex networks. *Computing*. 2019;101:1147–1175.
- Chen W, Lakshmanan LV, Castillo C. Information and influence propagation in social networks. *Synth Lect Data Manag*. 2013;5:1–177.
- Goyal A, Lu W, Lakshmanan LV. SIMPATH: An efficient algorithm for influence maximization under the linear threshold model. In: 2011 IEEE 11th International Conference on Data Mining. Vancouver, Canada: IEEE, 2011. pp. 211–220.
- Kempe D, Kleinberg J, 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. Washington, DC: ACM, 2003. pp. 137–146.
- Domingos P, Richardson M. Mining the network value of customers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, CA: ACM, 2001. pp. 57–66.
- Ju W, Chen L, Li B, et al. A new algorithm for positive influence maximization in signed networks. *Inf Sci*. 2020;512:1571–1591.
- Keikha MM, Rahgozar M, Asadpour M, et al. Influence maximization across heterogeneous interconnected networks based on deep learning. *Expert Syst Appl*. 2020;140:112905.
- Li Y, Fan J, Wang Y, et al. Influence maximization on social graphs: A survey. *IEEE Trans Knowl Data Eng*. 2018;30:1852–1872.
- Tong G, Wu W, Tang S, et al. Adaptive influence maximization in dynamic social networks. *IEEE ACM Trans Netw*. 2016;25:112–125.
- Wang Y, Fan Q, Li Y, Tan K-L. Real-time influence maximization on dynamic social streams. *Proc VLDB Endow*. 2017;10:805–816.
- Yang Y, Mao X, Pei J, et al. Continuous influence maximization. *ACM Trans Knowl Discov Data*. 2020;14:1–38.
- Beni HA, Bouyer A. TI-SC: top-k influential nodes selection based on community detection and scoring criteria in social networks. *J Ambient Intell Human Comput*. 2020;11:4889–4908.
- Güney E. On the optimal solution of budgeted influence maximization problem in social networks. *Oper Res*. 2019;19:817–831.
- Liu W, Chen X, Jeon B, et al. Influence maximization on signed networks under independent cascade model. *Appl Intell*. 2019;49: 912–928.
- Singh SS, Kumar A, Singh K, et al. IM-SSO: Maximizing influence in social networks using social spider optimization. *Concurr Comput*. 2020;32: e5421.
- Huang H, Shen H, Meng Z, et al. Community-based influence maximization for viral marketing. *Appl Intell*. 2019;49:2137–2150.
- Jing D, Liu T. Context-based influence maximization with privacy protection in social networks. *EURASIP J Wirel Commun Netw*. 2019;2019: 142.
- Rui X, Yang X, Fan J, et al. A neighbour scale fixed approach for influence maximization in social networks. *Computing*. 2020;102: 427–449.
- Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose, CA: Association for Computing Machinery, 2007. pp. 420–429.
- Wu P, Pan L. Scalable influence blocking maximization in social networks under competitive independent cascade models. *Comput Netw*. 2017; 123:38–50.
- Chen W, Lin T, Tan Z, et al. Robust influence maximization. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, CA: ACM, 2016. pp. 795–804.
- Huang K, Wang S, Bevilacqua G, et al. Revisiting the stop-and-stare algorithms for influence maximization. *Proceedings VLDB Endowment*. 2017;10:913–924.
- Shang J, Zhou S, Li X, et al. CoFIM: A community-based framework for influence maximization on large-scale networks. *Knowl Based Syst*. 2017;117:88–100.
- Gong M, Yan J, Shen B, et al. Influence maximization in social networks based on discrete particle swarm optimization. *Inf Sci*. 2016;367:600–614.
- Pei S, Teng X, Shaman J, et al. Efficient collective influence maximization in cascading processes with first-order transitions. *Sci Rep*. 2017;7:45240.
- Lu F, Zhang W, Shao L, et al. Scalable influence maximization under independent cascade model. *J Netw Comput Appl*. 2017;86:15–23.
- Peng S, Zhou Y, Cao L, et al. Influence analysis in social networks: A survey. *J Netw Comput Appl*. 2018;106:17–32.
- Banerjee S, Jenamani M, Pratihari DK. A survey on influence maximization in a social network. *Knowl Inf Syst*. 2020;62:3417–3455.
- Maghami M, Sukthankar G. Hierarchical influence maximization for advertising in multi-agent markets. In: 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013). Niagara, Ontario, Canada: IEEE, 2013. pp. 21–27.
- Kitsak M, Gallos LK, Havlin S, et al. Identification of influential spreaders in complex networks. *Nat Phys*. 2010;6:888.
- Narayanan R, Narahari Y. A shapley value-based approach to discover influential nodes in social networks. *IEEE Trans Autom Sci Eng*. 2010;8: 130–147.
- Goyal A, Bonchi F, Lakshmanan LVS. A data-based approach to social influence maximization. *Proc VLDB Endow*. 2011;5:73–84.
- Wang C, Chen W, Wang Y. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Min Knowl Discov*. 2012;25:545–576.
- Jung K, Heo W, Chen W. IRIE: Scalable and robust influence maximization in social networks. In: 2012 IEEE 12th International Conference on Data Mining. Brussels, Belgium: IEEE, 2012. pp. 918–923.
- Chen W, Lu W, Zhang N. Time-critical influence maximization in social networks with time-delayed diffusion process. In: Twenty-Sixth AAAI Conference on Artificial Intelligence. Toronto, Canada: AAAI Press, 2012. pp. 592–598.
- Cheng S, Shen H, Huang J, et al. Staticgreedy: Solving the scalability-accuracy dilemma in influence maximization. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management. San Francisco, CA: ACM, 2013. pp. 509–518.

39. Morone F, Min B, Bo L, et al. Collective influence algorithm to find influencers via optimal percolation in massively large social media. *Sci Rep*. 2016;6:30062.
40. Zhang J-X, Chen D-B, Dong Q, et al. Identifying a set of influential spreaders in complex networks. *Sci Rep*. 2016;6:27823.
41. Nguyen D-L, Nguyen T-H, Do T-H, et al. Probability-based multi-hop diffusion method for influence maximization in social networks. *Wirel Pers Commun*. 2017;93:903–916.
42. Zareie A, Sheikahmadi A. A hierarchical approach for influential node ranking in complex social networks. *Expert Syst Appl*. 2018;93:200–211.
43. Sun Y, Ma L, Zeng A, et al. Spreading to localized targets in complex networks. *Sci Rep*. 2016;6:38865.
44. Liu D, Jing Y, Zhao J, et al. A fast and efficient algorithm for mining top-k nodes in complex networks. *Sci Rep*. 2017;7:43330.
45. Li L, Li K, Xiang C. A hierarchy based influence maximization algorithm in social networks. In: *International Conference on Artificial Neural Networks*. Rhodes, Greece: Springer, 2018. pp. 434–443.
46. Xin Y, Gao C, Wang Z, et al. Discerning influential spreaders in complex networks by accounting the spreading heterogeneity of the nodes. *IEEE Access*. 2019;7:92070–92078.
47. Talukder A, Alam MGR, Tran NH, et al. Knapsack-based reverse influence maximization for target marketing in social networks. *IEEE Access*. 2019;7:44182–44198.
48. Qiu L, Jia W, Yu J, et al. PHG: A three-phase algorithm for influence maximization based on community structure. *IEEE Access*. 2019;7:62511–62522.
49. Ding J, Sun W, Wu J, et al. Influence maximization based on the realistic independent cascade model. *Knowl Based Syst*. 2020;191:105265.
50. Aghaee Z, Kianian S. Influence maximization algorithm based on reducing search space in the social networks. *SN Appl Sci*. 2020;2:1–14.
51. Kunegis J. KONECT: The koblenz network collection. In: *Proceedings of the 22nd International Conference on World Wide Web*. Rio de Janeiro, Brazil: Association for Computing Machinery, 2013. pp. 1343–1350.
52. Boguná M, Pastor-Satorras R, Díaz-Guilera A, et al. Models of social networks based on social distance attachment. *Phys Rev E*. 2004;70:056122.
53. Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters. *ACM Trans Knowl Discov Data*. 2007;1:2-es.
54. Zafarani R, Liu H. Social computing data repository at ASU. 2009.
55. Barabási A-L, Albert R. Emergence of scaling in random networks. *Science*. 1999;286:509–512.

**Cite this article as:** Beni HA, Bouyer A (2021) Identifying influential nodes using a shell-based ranking and filtering method in social networks. *Big Data* 9:3, 219–232, DOI: 10.1089/big.2020.0259.

### Abbreviations Used

CI = collective influence  
GIN = group of influential nodes  
HBIM = hierarchy based influence maximization  
IC = independent cascade  
IM = influence maximization  
IMP = influence maximization problem  
LIR = local index rank  
LT = linear threshold  
MIA = maximum influence arborescence  
NP = nondeterministic polynomial  
PHG = partition-heuristic-greedy  
RIC = realistic independent cascade  
RLP = reversed local path  
ROI = range of influence  
SPIN = ShaPley value based influential nodes  
SRFM = shell-based ranking and filtering method  
TI-SC = Top-k Influential nodes based on Scoring criteria and Community detection