Expert Systems With Applications 213 (2023) 118869

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

FIP: A fast overlapping community-based influence maximization algorithm using probability coefficient of global diffusion in social networks

Asgarali Bouyer [a,*,1], Hamid Ahmadi Beni [a], Bahman Arasteh [b], Zahra Aghaee [c], Reza Ghanbarzadeh [d]

[a] Department of Software Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran
[b] İstinye University, İstanbul, Turkey
[c] University of Isfahan, Isfahan, Iran
[d] Faculty of Science and Engineering, Southern Cross University, Gold Coast, Australia

ARTICLE INFO

Keywords:
Influence maximization
Overlapping nodes
Probability coefficient of global diffusion
Social networks
Community detection

ABSTRACT

Influence maximization is the process of identifying a small set of influential nodes from a complex network to maximize the number of activation nodes. Due to the critical issues such as accuracy, stability, and time complexity in selecting the seed set, many studies and algorithms has been proposed in recent decade. However, most of the influence maximization algorithms run into major challenges such as the lack of optimal seed nodes selection, unsuitable influence spread, and high time complexity. In this paper intends to solve the mentioned challenges, by decreasing the search space to reduce the time complexity. Furthermore, It selects the seed nodes with more optimal influence spread concerning the characteristics of a community structure, diffusion capability of overlapped and hub nodes within and between communities, and the probability coefficient of global diffusion. The proposed algorithm, called the FIP algorithm, primarily detects the overlapping communities, weighs the communities, and analyzes the emotional relationships of the community's nodes. Moreover, the search space for choosing the seed nodes is limited by removing insignificant communities. Then, the candidate nodes are generated using the effect of the probability of global diffusion. Finally, the role of important nodes and the diffusion impact of overlapping nodes in the communities are measured to select the final seed nodes. Experimental results in real-world and synthetic networks indicate that the proposed FIP algorithm has significantly outperformed other algorithms in terms of efficiency and runtime.

1. Introduction

With the development of information technology, social networks are growingly used. Social networks are considered as a server to share thoughts, news, and any information. People extensively and easily interact with each other through networks. An individual's behavior, such as buying a particular product, may affect others due to more relationships and collaboration with others in social networks (Berahmand, Bouyer, & Samadi, 2018; Bouyer, Azad, & Rouhi, 2022; Singh, Kumar, Singh, & Biswas, 2019). Therefore, social networks have become a powerful tool to expand the information in today's world, and it is an important incoming source for corporations and online marketing. For example, a company can create a wide advertising cascade on social media by selecting some people to use their products for free. Hence, the company must select users who have the greatest effect for accepting this product and recommend it to others. Therefore, it is seen that selecting the most influential people with high information cascades on social networks is an important problem. The influence maximization problem is proposed as an approach to choose influential people that can help the companies' commercial, improvement of the recommender systems, rumor diffusion control, identify the targets to immunize or quarantine for preventing an epidemic in a population(Cherifi, Palla,

* Corresponding author.
E-mail addresses: a.bouyer@azaruniv.ac.ir (A. Bouyer), h.ahmadi@azaruniv.ac.ir (H. Ahmadi Beni), Bahman.arasteh@istinye.edu.tr (B. Arasteh), Z.Aghaee@eng.ui.ac.ir (Z. Aghaee), Reza.Ghanbarzadeh@scu.edu.au (R. Ghanbarzadeh).
[1] https://orcid.org/0000-0002-4808-2856.

https://doi.org/10.1016/j.eswa.2022.118869
Received 6 May 2022; Received in revised form 13 September 2022; Accepted 17 September 2022
Available online 24 September 2022

Szymanski, & Lu, 2019; Gong et al., 2013), network monitoring, and so forth.

Several researchers have attracted the issue of influence maximization in recent years (Aghaee, Beni, Kianian, & Vahidipour, 2020; Kazemzadeh, Karian, Safaei, & Mirzarezaee, 2021). Notably, Domingos and Richardson have first introduced the influence maximization problem (Domingos & Richardson, 2001). Then, Kempe et al. developed and formulated this problem and demonstrated that it is an NP-hard problem(Kempe, Kleinberg, & Tardos, 2003). Therefore, the influence maximization problem refers to the selection of k minimum nodes that have the highest influence spread in social networks. Likewise, the independent cascade model and linear threshold limit are the two popular models in the influence maximization problem. Diffusion probability is an important parameter in the diffusion models that indicates the probability of a node's effect on another node. Furthermore, the influence maximization problem has sub-modularity and monotonous properties in the independent cascade and linear threshold models (W. Chen, Wang, & Yang, 2009).

Recently, researchers examined the influence maximization problem based on the community detection approach. Recently, many local community detection algorithms with linear time complexity (Aghaalizadeh, Afshord, Bouyer, & Anari, 2021; Bouyer & Roghani, 2020; Zarezadeh, Nourani, & Bouyer, 2021) have been proposed that are used as a preprocessing step in IM problem. These algorithms are more efficient than the greedy algorithm because the calculation of the influence spread is limited to the communities (Aghaee, Ghasemi, Beni, Bouyer, & Fatemi, 2021). However, community-based detection algorithms have several drawbacks: 1. these algorithms cannot examine the probability of global diffusion with regard to the structure of the communities, while the investigation of the probability of global diffusion in the communities must be measured as an important indicator. 2. The algorithms do not deal with any approach to diminish the search space for selecting the seed nodes, whereas the search space should be decreased to improve the efficiency in the large-scale social networks 3. They do not address the communities' topological characteristics to calculate the influence spread, while suitable communities for influence spread are detected by examining the communities' topological characteristics. 4. Most of these algorithms examine the nodes with the core-role to select the final seed nodes, whereas the role of other nodes such as hub or bridge nodes is not examined in the optimum selection of the seed nodes (Samadi & Bouyer, 2019). For example, the PHG algorithm can be very time-consuming on large-scale networks despite its simple nature. Since the algorithm must check all communities at each iteration, the running time increases due to the large search space. Therefore, the FIP algorithm is proposed to solve the mentioned drawbacks of community-based influence maximization under the independent cascade model. The new algorithm is more efficient in runtime and influence spread than the algorithms developed in recent years. This method is based on the detection of overlapping communities and includes two general phases. In the first phase, the LPANNI algorithm (Lu, Zhang, Qu, & Kang, 2018)is used to detect overlapping communities. Each community's weight is determined by the topological characteristics of the communities and analysis of the community nodes' emotional relationships. At the next, the search space is limited by computing θ_c criterion, which removes the insignificant communities. In the second phase, candidate nodes are selected by examining the diffusion probability of intra-community and coefficient of diffusion probability of outer-community. Then, the seed nodes are selected from candidate set nodes and the best overlapping nodes. To sum up, our major contributions in this paper are:

1. FIP algorithm is proposed based on the overlapping communities for an influence maximization problem. The search space is limited in the communities' filtering phase to improve the runtime efficacy. Consequently, the algorithm is applicable to large-scale networks.

2. This algorithm uses the analysis of the node's emotional relationships to examine the communities suitable for the influence spread according to social interactions' power. This step has a required effect on the quality of the seed nodes selection.

3. According to the FIP algorithm, the seed nodes are chosen with regard to the structure of neighbors and the role of the nodes in communities. Sometimes, overlapping nodes may have a critical role in the spreading of the influence. For this reason, this algorithm significantly has higher accuracy in selecting the seed nodes at a very low frequency in the Monte Carlo simulation, whereas the other existing methods have very low accuracy in selecting the seed nodes at the low frequency of the Monte Carlo simulations.

4. We perform comprehensive tests on social networks' real and synthetic datasets. The obtained results show the efficiency of the FIP algorithm in comparison to other compared methods.

The rest of the paper is organized as follows. Section 2 contains a valuable review of the literature. Section 3 provides a detailed description of the proposed method. Section 4 deals with the experimental evaluation for the proposed algorithm, and Section 5 draws the conclusion of the present study.

2. Related work

With the advance in social networks, many scholars focus on the issue of influence maximization. The related studies in this topic are classified into two categories:

1. Diffusion-based algorithms
2. Heuristic algorithms

These categorized algorithms are briefly explained in subsections 2.1 and 2.2.

2.1. Algorithms based on diffusion

For the first time, influence maximization was proposed by Domingos and Richardson as an algorithmic method(Domingos & Richardson, 2001). Later on, Kempe developed and formulated influence maximization (Kempe et al., 2003). They developed the Greedy algorithm, which calculates each node's influence spread using the Monte Carlo simulation (Kempe et al., 2003). The Greedy algorithm guarantees the optimal approximation, but because the Monte Carlo simulation requires a high number of iterations for the greedy algorithm, it is inefficient on large-scale networks. Consequently, Leskovec et al. Developed the CELF algorithm, which uses lazy evaluation to reduce the influence spread calculations (Leskovec, Krause, et al., 2007). However, the CELF algorithms were still inefficient due to the use of Monte Carlo simulations in the high number of iterations. Chen et al. developed the NewGreedyIC algorithm, which calculates the influence spread for every node using a set of available nodes. Also, Chang et al. proposed the StaticGreedyDU algorithm to improve the NewGreedyIC algorithm's runtime (Cheng, Shen, Huang, Zhang, & Cheng, 2013). However, the StaticGreedyDU algorithm does not require efficacy in finding the optimal seeds. Goyal et al. developed a SIMPATH algorithm that selects seed nodes by limiting simple path counts [19] to speed up the influence spread computation. The SIMPATH algorithm has an excellent runtime for choosing K node seed and low memory overhead but does not guarantee optimal approximation. In addition, the SRFM was proposed as a shell-based ranking and filtering method, for selecting optimal seed set with the aim to maximize influence in a near-linear time complexity (Ahmadi Beni & Bouyer, 2021). It initially filters periphery nodes and some nodes from unimportant shells. The MIA algorithm was proposed by Wang et al., which is a scalable algorithm, and the influence spread is calculated locally using an arborescence tree (C. Wang, Chen, & Wang, 2012). The MIA algorithm has a good runtime because it performs influence spread computation without using Monte Carlo simulation. Another weakness of this algorithm is its' highly overhead memory.

2.2. Heusteic algorithms

Chen et al. proposed a DegreeDiscount algorithm to improve the running time of the NewGreedyIC algorithm, which initially selects a node with the highest degree as the first node of the seed. It then selects seed nodes by decreasing the degree of nodes that have seeds in the neighborhood (W. Chen et al., 2009). The DegreeDiscount algorithm does not guarantee the optimal approximation, but it has an excellent runtime compared to the Greedy, CELF, and NewGreedyIC algorithms. Then, Narayanam et al. developed the SPIN algorithm, which uses the Shapley value to calculate the influence spread for each node (Narayanam & Narahari, 2010). This algorithm has a good runtime but does not guarantee optimal approximation. Then, Kitsak et al. developed a K-core algorithm that specifies the core and periphery nodes in the graph and considers the core nodes as seed nodes (Kitsak et al., 2010). Zhang et al. developed the VoteRank algorithm based on voting (J.-X. Zhang, Chen, Dong, & Zhao, 2016). This algorithm also does not have proper quality in selecting the seed nodes due to mainly focus on fast run time. In addition, this algorithm does not have submodularity. Then, Chen et al. developed a CIM algorithm based on community detection (Y.-C. Chen, Zhu, Peng, Lee, & Lee, 2014). Although the CIM algorithm has a good runtime, it has a problem in identifying suitable seed nodes. Then, Shang et al. developed the CoFIM algorithm, which also is a community-based approach (Shang, Zhou, Li, Liu, & Wu, 2017). Its performance was better than CIM. However, the runtime of the CoFIM algorithm depends on the number of seed nodes. At that time, Morone et al. proposed the CI algorithm based on localization of influence spread computation (Morone, Min, Bo, Mari, & Makse, 2016). In this algorithm, the influence spread computation in a circle is limited to the radius L, but the runtime of this algorithm depends on L and the number of seed nodes. Also, Liu et al. developed the LIR algorithm based on heuristic methods (Liu, Jing, Zhao, Wang, & Song, 2017). In the LIR algorithm, the LI value for each node is computed according to the degree of neighbors, and then, the set of nodes with the lowest LI value is sorted in descending order, and k nodes are selected as seeds. Furthermore, the LMP algorithm proposed as a local and fast method for identifying influential nodes in a linear-time complexity (Bouyer & Beni, 2022). It primarily reduces the search space using a node labeling approach. In the LMP algorithm, after ranking the nodes in a local and fast step, the nodes with the highest ranking-label are selected as candidate nodes. Then, the final seed set is identified using the topology features and the strategic position of the candidate connector nodes. Nguyen et al. developed the ProbDeg that uses multi-hop neighbors and propagation probabilities of nodes to select seed nodes (Nguyen, Nguyen, Do, & Yoo, 2017). This algorithm does not provide an approximation guarantee. Ahajjam et al. developed the HybridRank algorithm, which selected seed nodes based on eigenvector centrality and coreness (Ahajjam & Badir, 2018). In this algorithm, the selection of seed nodes is avoided by the Rich-club phenomenon. Saxena et al. developed a novel centrality measure social centrality based on an individual's propensity to socialize and aggregated to produce social centrality score (Saxena, Kaur, & Bhatnagar, 2018). Wu et al., Proposed the LAIM as a linear time iterative approach for the influence maximization problem on large-scale networks(Wu et al., 2018). This algorithm has a low memory overhead on large-scale social networks. Banerjee et al. developed a ComBIM algorithm that selects seed nodes according to the community budget (Banerjee, Jenamani, & Pratihar, 2019). This algorithm does not provide an approximation guarantee.

Li et al. (W. Li, Zhong, Wang, & Chen, 2021) used a community overlap propagation algorithm based on cohesive entropy (CECOPA) for overlapping community detection in networks. Then, they explored the dynamics of propagation and the influence of local aggregation factors on influence diffusion, and proposed a dynamic influence maximization algorithm based on cohesive entropy using Optional Dynamic influence Propagation algorithm (ODP). Also, Xie et al. proposed the IRR algorithm for the MBIC model (Xie, Chen, Zhang, & Liu, 2019). This algorithm divides the influence propagation process into two stages, the influence stage and the reference stage. This algorithm has a better influence spread than the DegreeDiscount algorithm. Rui et al. developed the RNR algorithm (Rui, Meng, Wang, & Yuan, 2019) that calculates each node's influence power using neighbors. Qiu et al. developed a PHG algorithm that uses a community-based approach. It also uses the greedy algorithm to influence spread computation(Qiu, Jia, Yu, Fan, & Gao, 2019). However, the influence spread and runtime depend on the number of seed nodes selected by the greedy algorithm. Ghalmane et al. developed a Modular centrality that this centrality has a two-dimensional vector (Ghalmane, El Hassouni, Cherifi, & Cherifi, 2019). The two-dimensional vector is based on the local influence of a node in its community and global influence on the other communities of the network. Lin et al. improved an activation effect based on a hybrid distribution value accumulation algorithm in linear threshold model that has 2 stages, value greed stage and mountain climbing algorithm stage (Lin, Zhang, Xia, Ren, & Li, 2019). Wang et al. proposed a new algorithm to calculates influence spread that in independent cascade model defined the probability of the node attraction. As well as this, the probability of the node attraction was designed by the node properties (G. Wang, Jiang, Li, & Wang, 2019). Ghalmane et al. proposed an Overlapping Modular Centrality to select the influential nodes using the local and global influence of overlapping and non-overlapping nodes (Ghalmane, Cherifi, Cherifi, & El Hassouni, 2019). Ahmadi Beni et al. developed a community-based algorithm to examine the relationships between the core nodes and the scoring ability of other nodes in communities (Beni & Bouyer, 2020). Li et al. proposed a new model for the propagation speed in information diffusion (W. Li et al., 2020). The idea is based on the fact that by moving away from the source in social networks, the spread becomes weaker. Chakrapani et al. proposed a new problem in the influence maximization problem that this problem is based on the time sensitive. Also, seed nodes are selected within the specified time limit (Chakrapani, Chourasia, Gupta, & Haldar, 2021). Li et al. proposed a Gaussian propagation model instead of an independent cascade model and linear threshold model; that this algorithm improves the CELF algorithm based on submodularity (W. Li, Li, Luvembe, & Yang, 2021). This algorithm is not fast in networks with a large number of communities. Kazemzadeh et al. proposed the CTIM algorithm that there is a positive correlation between influential nodes and high charismatic power in this algorithm. This algorithm uses local and global diffusion to select seed nodes (Kazemzadeh, Safaei, & Mirzarezaee, 2022). Zhang et al. proposed the GCNM algorithm based on the network dynamic GCN with aim to adjusting the size of social networks and the size of neural networks (C. Zhang, Li, Wei, Liu, & Li, 2022). They also implemented a leader fake labeling mechanism to generate a node label for each node for model training. At the end, a set of nodes with more considerable influence is identified as the seed set based on the learned node representations.

Li et al. proposed the MAHE-IM algorithm based on a multiple aggregation of heterogeneous relation embedding to identify seed nodes that influence spread calculated by the global and local features in the heterogeneous information network (Wei, Zhao, Liu, & Wang, 2022). Li et al. defined text emotion-power for selection influential node based on the group dynamics theory that behavior is defined as the result of interaction between needs and forces. The TFG algorithm does not select optimal seed nodes because Greedy and Degree have better influence spread than the TFG algorithm (W. Li, Li, Liu, & Wang, 2022). The LMP algorithm proposed to select influential seed nodes that assigned labels for each node (Bouyer & Beni, 2022). Feng et al. proposed a new method by considering fine-grained discounts and assume users accept the discount probabilistically (Feng et al., 2022). Then, they defined and proved a new concept in their method that is called the set-wise friendship paradox (FP) phenomenon. Chaharborj et al. proposed the TOPSIS method to detect seed sets that used the transmission probability of an epidemic disease based on the Susceptible-infected-recovered (SIR) epidemic model. Their finding indicates that influential nodes

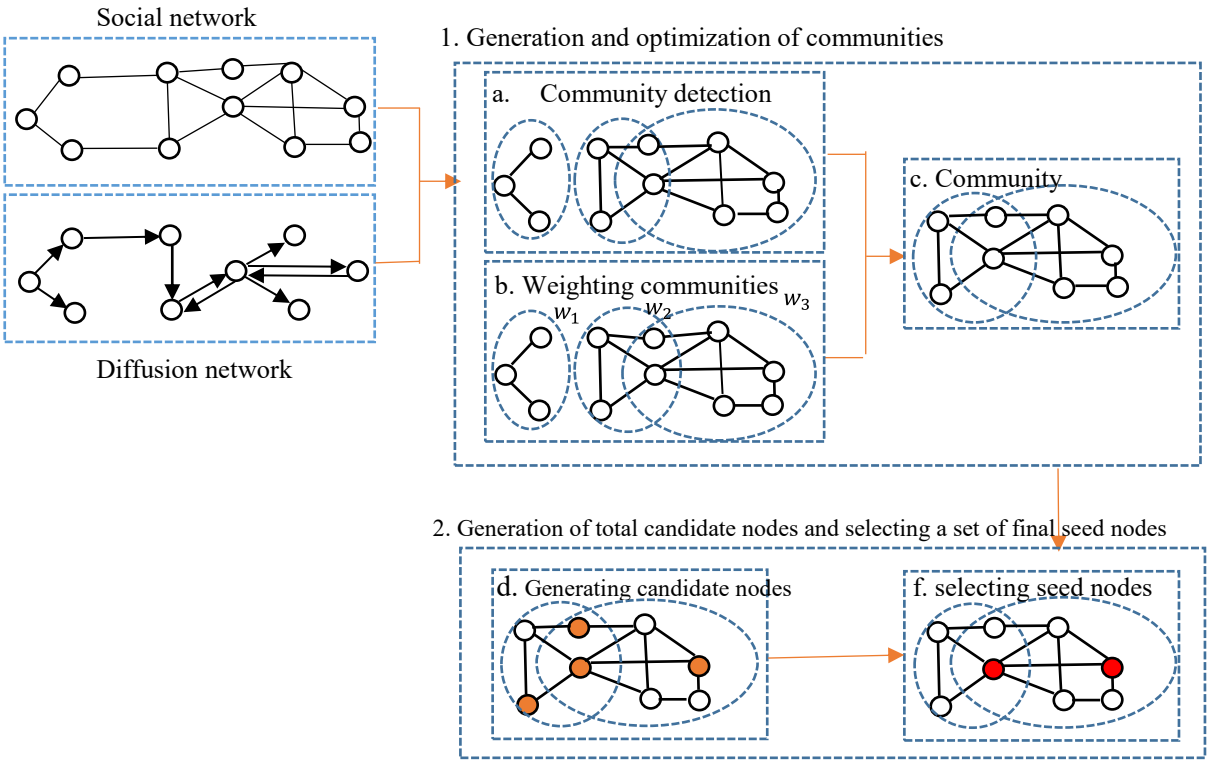


Fig. 1. A typical example for steps of the FIP algorithm.

have absolutely crucial role in curbing the diffusion of COVID-19 (Chaharborj, Nabi, Feng, Chaharborj, & Phang, 2022). Li et al. proposed a new gravity model to identify influential spreaders based on different features like k-shell value and eigenvector centrality value. The influence spread in this algorithm depends on truncation radius, which is based on the neighborhood criterion of nodes (Z. Li & Huang, 2022). Wang et al. proposed the LENC algorithm that calculates the information entropy value by the edge weight value of nodes for the selection of influential nodes. In addition, in this algorithm, the diffusion of first-level and second-level edges of the nodes are considered (B. Wang, Zhang, Dai, & Sheng, 2022). Wei et al. proposed a new method for finding influential nodes for target immunization strategies that they found that betweenness, degree, H-index, and Coreness have an important role in the disease spreading model in scale-free networks (Wei et al., 2022). Kumar et al. proposed the CSR algorithm that detects influential nodes based on community diversity, community modularity, and community density that the community diversity depends on its edges with different communities (Kumar, Gupta, & Khatri, 2022). Zhang et al. proposed the IM-NM approach to find k influential nodes based on high important network motifs (X. Zhang, Xu, & Xu, 2022). In this method, the high important network motifs are discovered by defining structural stability level, weight ratio, and degree density concepts. At next, the Naive Bayes algorithm is used to classify high importance motif. Finally, the k-influential nodes with best bridge and strong communication capability are identified from selected key motifs. Despite of its high computational time, its infection capability is not satisfactory in comparison to local methods such as degree centrality or global methods such as BC or CC centralities.

3. The proposed method

FIP algorithm is a community-based detection method for the influence maximization problem, which effectively chooses the influential nodes using the communities weighing and the probability of global diffusion. The FIP algorithm includes two main steps: 1. Generation and optimization of communities; and 2. Generation a set of candidate nodes and select the final seed set. The steps of the FIP algorithm are depicted in Fig. 1. In Fig. 1.a communities discover based on LPANNI. Then, In Fig. 1.b, communities' weights are calculated. After that, in Fig. 1.c, unsuitable communities for influence spread are ignored optimally. Then, in Fig. 1.d, candidate nodes are selected based on the probability coefficient of community diffusion. Finally, In Fig. 1.f, nodes select from the best non-overlapping influential nodes and importance overlapping nodes as seed nodes.

3.1. Community detection and reduction

Community structure is one of the main properties of social networks. In these networks, nodes are often connected within the cluster or community with high density, while the density of the communication between clusters is low (Roghani & Bouyer, 2022; Roghani, Bouyer, & Nourani, 2021; Taheri & Bouyer, 2020). In the influence maximization problem, discovering community structures help us to reduce computational overhead. In our proposed algorithm, the LPANNI algorithm is used for community detection, which is a well-known algorithm to detect communities with overlapping (Lu et al., 2018). The LPANNI algorithm detects communities based on node importance, label update strategy and historical label preferred strategy that has two vitally important features: low time complexity and stability of community detection. After the LPANNI algorithm detects the communities, overlapping nodes in the communities are identified. Then, the communities are weighted based on equation (1). It needs to compute each community's weight to understand whether the community is suitable for influence calculations. According to our experiments, many communities are not suitable for influence calculations. Some communities are suitable for influence spread because nodes in those communities can spread information to other communities. Regarding the structure of discovered communities, some unimportant communities must be filtered to avoid seed node selection from these communities. Therefore, it is not necessary to calculate the influence spread for all nodes and

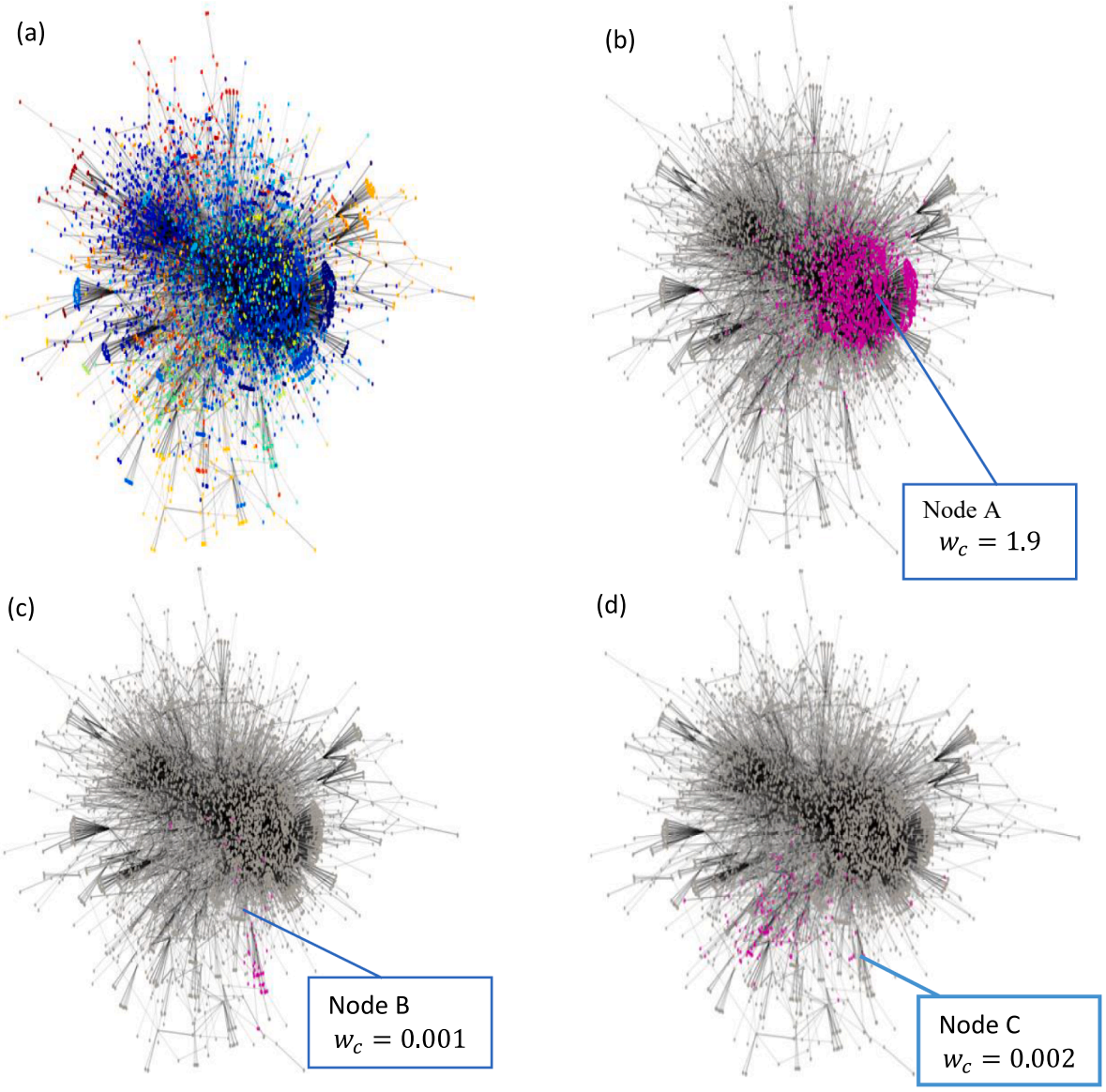


Fig. 2. The amount of diffusion in communities with different w_c. (a) indicates the *Route views* network. Each color represents the community. (b) shows the amount of diffusion for node A in a community with $w_c = 1.9$. The blue line refers to the node that is activated in the community to spread information on the network. The pink nodes depict the amount of diffusion for node A. (c) represents the amount of diffusion for node *B* with $w_c = 0.001$. The blue line refers to the node that is activated in the community to spread information on the network. Also, pink nodes show the amount of diffusion for node *B*. (d) represents the amount of diffusion for node *C* with $w_c = 0.002$. The blue line refers to the node that is activated in the community to spread information on the network. The pink nodes show the amount of diffusion for this node.

communities. The topological structure of the societies and the emotional dependence of the nodes within communities are important for determining the spread of influence. Hence, some communities must effectively be removed to optimally reduce the search space and select the most influential seed nodes. Therefore, for each graph $G=(V, E)$, $C = \{c_1, c_2, c_3, \cdots, c_l\}$ is detected communities where $c_i = (v_c, e_c)$ so that $v_c \in V$ and $e_c \in E$, and $w_c = \{w_{c1}, w_{c2}, w_{c3}, \cdots, w_{cl}\}$ are the computed weights of communities. The weight of a community C_i is defined as follows (w_{ci}):

$$w_{ci} = \left(\frac{e_{ci} + e_{oi}}{e}\right)\frac{n_{ci}}{n} + \left(log\left(\sum_{i=1}^{v_{ci}} \frac{dis_{v_{ci}}}{deg_{v_{ci}}} + 1\right) + as\right)\frac{n_{ci}}{n} \quad (1)$$

where e_{ci} is the edges within the community c_i, e_{oi} represents the number of edges between community c_i with other communities, and e is the total number of edges in the network. Moreover, n_{ci} is the number of the nodes in a community c_i and n is the total number of nodes. v_{ci} is a node in the community c_i and $deg_{v_{ci}}$ is its degree . In equation (1), a new network-based characterization is presented, which identifies the romantic feelings of two people based on the structure and characteristics of the network. Also, in principle, we do not only extract emotional characteristics in equation (1). If the amount of dispersion is high, two nodes have a strong emotional relationship, if two nodes have less dispersion, two nodes have a weak emotional relationship.

Also, *as* is *assortativity coefficient* is defined for indirect networks that is defined by equation (2) as follow (Newman, 2003):

$$as = \frac{\sum_{i,j}^{n_{ci}} (deg_{v_{ci}} - \overline{deg_{v_{ci}}})(deg_{v_{cj}} - \overline{deg_{v_{cj}}})}{\sqrt{\sum_{i}^{n_{ci}} (deg_{v_{ci}} - \overline{deg_{v_{ci}}})}\sqrt{\sum_{j}^{n_{ci}} (deg_{v_{cj}} - \overline{deg_{v_{cj}}})}} \quad (2)$$

where $\overline{deg_{v_{ci}}}$ and $\overline{deg_{v_{cj}}}$ are the average degree of nodes v_{ci} as sources, and the average degree of nodes v_{cj} as targets, respectively. The higher the *as* a criterion, the more similar nodes in the community are connected, so

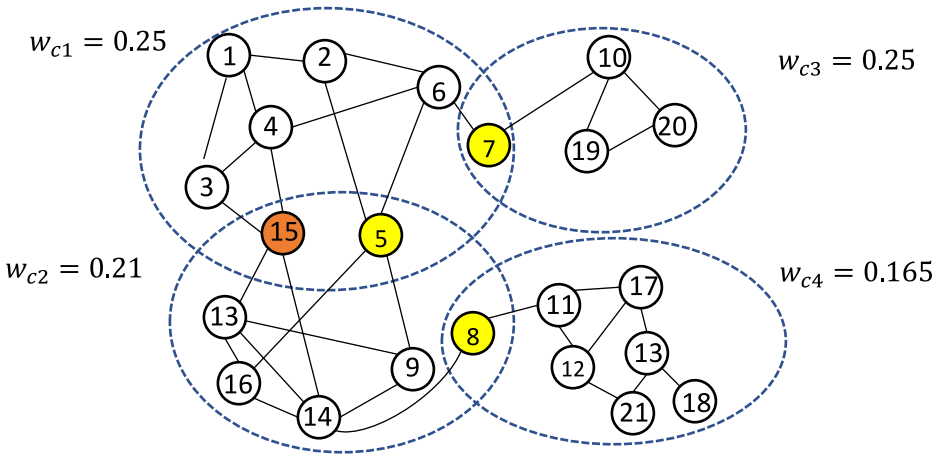


Fig. 3. An example for reducing of communities.

the dissemination of information in this community can be optimal. Furthermore, $dis_{v_{ci}}$ represents the *dispersion* criterion to check the amount of emotional relationships in the community c_i (Backstrom & Kleinberg, 2014). It should be noted that the sub-graph $G_{v_{ci}}$ is achieved in the analysis of emotional relationships for each node v_{ci} that includes the neighbors of the node v_{ci}. For each node u in the sub-graph $G_{v_{ci}}$($G_{v_{ci}}$ is node v_{ci} and all neighbors of v_{ci}), the set $C_{uv_{ci}}$ are all common neighbors between v_{ci} and each node u in its sub-graph $G_{v_{ci}}$. Therefore, according to equation (3), *dispersion* is calculated according to each pair node in $C_{uv_{ci}}$ set, where d_v is a distance function for nodes of $C_{uv_{ci}}$ and s and t are two nodes in $C_{uv_{ci}}$. The distance function d_v is 1 when s and t are not directly connected and also do not have common neighbors in sub-graph $G_{v_{ci}}$, and equal to 0 otherwise (Backstrom & Kleinberg, 2014).

$$dis(v_{ci}) = \sum_{s,t \in C_{uv_{ci}} \& s \neq t} d_v(s,t) \tag{3}$$

The higher the *as* a criterion refers to the closeness of a friendship, so the dissemination of information in this community can be optimal. The first part of equation (1) examines the topological criterion and density of the nodes and edges for a community c_i. The second part indicates the degree of emotional relationships and dependence of the nodes of the community. In equation (1), the number of nodes and edges cannot be a good feature for selecting an optimal community. As a result, communities that have nodes with the most emotional relationship and similarity select as optimal communities for diffusion.

After discovering communities by the CPM algorithm, it is seen that the number of generated communities is high. On the other side, there are several weak communities with weak nodes to spread of the influence. Therefore, such insignificant communities should be eliminated in order to reduce the search space of the seed nodes selection. The main idea behind the community reduction is that the influence of weak communities without required emotional relationships and density cannot spread the influence on other communities. Therefore, to reduce some communities, the threshold θ_c is defined by equation (4) as follow:

$$\theta_c = den_{c,max} \times \frac{(n_{c,max} + k)}{A \times n_{c,min}}, \tag{4}$$

where $den_{c,max}$ is the density of the largest community. The higher value of $den_{c,max}$ means a great number of involved communities in the calculations of finding influential spreads. If the value of $den_{c,max}$ is low, a low number of communities is involved in the next step to explore the influential nodes. Among the existing communities, $C = \{c_1, c_2, c_3, \cdots, c_l\}$., $n_{c,max}$ and $n_{c,min}$ represent the number of nodes in the largest and smallest communities, respectively. k refers to the number of seed nodes, and A stands for the number of detected communities. If $w_{ci} \geq \theta_c$, the community i can be suitable to explore and find influential nodes at a later step. It is possible that no seed node is selected in the next step. In other words, some of these selected communities do not necessarily have the best seeds.

Fig. 2 shows the amount of diffusion for three communities with different w_{ci}. In Fig. 2(a), each color represents a community. For example, nodes with blue color are in community 1. Also, yellow nodes are in community 2. In Fig. 2, the blue line refers to the node that is activated in the community to spread information on the network. This node in Fig. 2(b), Fig. 2(c), and Fig. 2(d) is node A, node B, and node C. In Fig. 2(b), pink nodes are nodes that are activated by node A. As seen in this figure, the number of the nodes activated in communities with high w_c is higher. Therefore, nodes within a community with higher w_{ci} can activate both nodes within their community as well as within other communities (Fig. 2 (b)). However, as shown in Fig. 2(c) and 2(d), the nodes in communities with low w_c can only activate a few nodes within their community, and thus diffusion in such communities is not spread to other nodes in other communities.

Algorithm 1: discovering and reducing communities of the network (G)

Input: Network G (V, E)
Output: Reduced Communities $C_b = \{c_{b1}, c_{b2}, \cdots, c_{bx}\}$
1: initialize $C \rightarrow \varnothing$ $C_b \rightarrow \varnothing$,;
2: Find initial communities $C = \{C_1.C_2 \cdots C_l\}$ by LPANNI algorithm. // step (i): Community detection
3: **for each** C_i in C **do** // step (ii): compute a weight for each Community
4: $w_{ci} = (\frac{e_c + e_o}{e}) \frac{n_c}{n} + (log(\sum_{i=1}^{v_c} \frac{dis}{N_{vc}} + 1) + as) \frac{n_c}{n}$
5: **end for**
6: **for each** c_i in C **do** // step (iii): Remove weak Communities
7: **if** $w_{ci} \geq \theta_c$ **then**
8: $C_b.add(c_i)$
9: **end if**
10: **end for**
11: return $C_b = \{c_{b1}, c_{b2}, \cdots, c_{bx}\}$

The output of algorithm 1 is the found suitable communities for diffusion. In line 2 of algorithm 1, the communities are discovered by the LPANNI algorithm, and then w_c is calculated for each community c_i in lines 3–5. Afterward, in lines 6–10, the unimportant communities are removed. For each community c_i, if $w_{ci} \geq \theta_c$ the c_i is added to the set C_b where C_b is a set of suitable communities for exploring the influential nodes.

Our proposed method for selecting overlap nodes completely is different. It only selects some specific and important overlapping nodes for the probability of selecting as candidate node. To prove the problem, pay attention to the example network in Fig. 3. In this example, we have 4 communities, the weight of each community is $w_{c1} = 0.25$, $w_{c1} = 0.21$, $w_{c1} = 0.151$, and $w_{c1} = 0.165$. Also, $\theta_c = 0.175$ to find suitable communities for influence spread. According to the w_c and θ_c criteria,

communities 1 and 2 are suitable for influence spread because $w_{c1} > \theta_c$ and $w_{c2} > \theta_c$. Also, communities 3 and 4 are ignored in the calculation of influence spread because $w_{c3} < \theta_c$ and $w_{c4} < \theta_c$. Community 2 and 4 have the same number of nodes, but the FIP algorithm chooses community 2 to influence the spread and selection of seed nodes. As can be seen, there are 4 overlapping nodes 15, 5, 7, and 8 in communities 1, 2, 3, and 4. The FIP algorithm selects the overlapping node 15 between community 1 and 2 as an important overlapping node as a candidate node because this node is located as a gatekeeper among the core nodes of the two communities, which plays a significant role in global diffusion process in the network. It is obvious that nodes 5 and 15 both have the same degree, but according to equation (7) in the FIP algorithm, node 15 has better strategic position than node 5 and it is added to the candidate set.

In the FIP algorithm, according to equations (6), 7, 10, and 11, the effect of some features such as node degree, the effect of the clustering coefficient of the node, and its neighbor nodes was evaluated for the overlap nodes. These topological features of overlap nodes have a significant effect on the influence spreading. It is worth noting that nodes with high clustering coefficient are located in dense regions of the graph and these nodes often have less effect in global diffusion. Because in these nodes, the number of common neighbors is very high, which necessarily causes local diffusion in the dense area and does not help to global diffusion in the network.

For example, suppose the faculty officials of a college intend to hold an international conference. The conference officials cannot have an optimal publication if they only spread the "call for paper" among the college staffs because they only propagate information repeatedly in a dense part of network due to large number of common neighbors for each node. To avoid this natural phenomenon in the real world, the FIP algorithm assign a particular importance for important overlaps in the strategic position, located among communities as a superhighway for information diffusion. But DEIM algorithm only consider the bridgeness of the overlap nodes that are connected to at least 6 communities whereas this selection procedure is not capable because many bridges have a low importance due to have connections with periphery nodes of different communities. Also, it does not care about the important features such as self-clustering coefficient and neighbor nodes' clustering coefficient. So, only some bridges with relatively high degree and low self-clustering coefficient and averagely higher clustering coefficient of their neighbors have a suitable strategic position. Therefore, it should be noted that the overlap nodes selected by the FIP algorithm play the important role in influence spreading because they mostly have connection with different community cores in the network, in addition to the mentioned characteristics.

3.2. Identifying influential candidate nodes

The main objective of the influence maximization problem is the optimum selection of the seed nodes. However, it should be noted that calculations of the influence spread for selecting the seed nodes are a time-consuming process. Therefore, to reduce the computational time, the FIP algorithm generates a limited number of candidate nodes for each selected community in the set C_b because the rest of the calculations of finding the final seed nodes are limited to the set of these candidate nodes. Moreover, according to the community's structure in social networks, the generating suitable candidate nodes and the probability coefficient of global diffusion play an important role in selecting the final seed nodes. Thus, each candidate node is examined as a potential seed node. Also, the candidate nodes are more important than the other nodes to be selected as the seed because these nodes are the bridge between several communities and can have the most significant influence on communities. Since the selection of candidate nodes depends on the probability coefficient of global diffusion in communities, the probability coefficient of global diffusion must first be examined for communities.

3.2.1. The probability coefficient of community diffusion

The *probability coefficient of community diffusion* has high importance in the influence maximization problem because for a node selected as a seed, the probability coefficient of community diffusion for such a node is higher than the other nodes. For calculating the probability coefficient of *community* diffusion, those nodes are considered that only have a connection inside the community. Therefore, the nodes of each community c_{ib} is divided into two sets v_I and v_O where $v_I = \{v_{I1}, v_{I2}, v_{I3}, \cdots, v_{Ih}\}$ is a set of nodes within the community c_{ib} that has no edge to other communities, and $v_O = \{v_{O1}, v_{O2}, v_{O3}, \cdots, v_{Oh}\}$ is a set of the nodes within a community that have some edges to other communities in the set C_b. Therefore, the probability coefficient of *community* diffusion for each node v_I is calculated by equation (5) as follow:

$$p_d = p_{ar}(n_{1i} + 2\sqrt{n_{2i}}) \tag{5}$$

where n_{1i} and n_{2i} respectively are the neighbors of the node v_i with distances 1 and 2 of the set v_O. In the real world, the probability of diffusion and reaching information to nodes in the set v_O is a random parameter function. For this reason, p_{ar} is considered as a random parameter, which is a random number with a uniform distribution between [0,1]. Finally, p_d values are normalized for all nodes. In general, p_d is a value between 0 and 1. When $p_d = 1$, the node v_i has the maximum probability of diffusion to other communities, while if $p_d = 0$, the node v_i has the minimum probability of global diffusion. Therefore, even though a node with low $p_{d,i}$ is directly not connected to other communities, but it may have a higher probability of global diffusion to other communities due to having more neighbor nodes in the v_O set at distances 1 and 2. Therefore, the node with high p_d can play an important role in diffusing information because of their close association with several bridge nodes that are connector among communities.

On the other hand, we calculate the probability coefficient of *community* diffusion for the nodes in the set v_o due to having an edge role between other communities. Therefore, for a node v_o within community c_{ib}, $\Gamma v_{co} = \{v_{co1}, v_{co2}, \cdots, v_{coj}\}$ are the direct neighbors of the node v_o in communities c_j. Furthermore, the set $v_{Hc} = \{v_{1_{HD}}, v_{2_{HD}}, v_{3_{HD}}, \cdots, v_{n_{HD}}\}$ is the node(s) with a maximum degree in each community c_j. It is necessary to mention that various communities may have nodes with the same maximum degree ($deg_{cj,max}$). All these nodes must be added to v_{Hc}. Thus, the probability coefficient of *community* diffusion for each node v_o is calculated as follows:

$$p_d = \frac{|v_{Hc}|.deg_{c,max}}{\sum_{ci=1}^{nc_c} deg_{(ci,max)}} \times \sum_{i=1}^{w} \frac{1}{(x_{sp_i} \times p_{ar_i})} \tag{6}$$

where $|v_{Hc}|$ is the number of the nodes with the maximum degree in the set v_{Hc}, w is the number of the neighbors of a node v_o, x_{sp_i} is the distance of the shortest path for each node v_{co_i} of Γv_{co} to its closest node in v_{Hc}, and n_c is the number of communities in C_b, $deg_{(ci,max)}$ is the maximum degree in community C_{b_i}, $deg_{c,max}$ is the maximum degree among all communities in C_b, and p_{ar_i} is the probability of accessing node v_{co_i} to other nodes in the diffusion process. In general, if $\Gamma v_{co} \in v_{Hc}$, it means that all neighbors of the node v_o are the nodes with the highest degree. In this case, the p_d has the maximum value. If $\Gamma v_{co} \cap v_{Hc} = \emptyset$, it means that none of the neighbors of the node v_o are not is the set v_{Hc}, and in this case, the node v_o has the lowest p_d value. In general, p_d is normalized to have a value between 0 and 1. If $p_d = 1$, the node v has the greatest global diffusion to other communities. Therefore, the main idea behind the probability coefficient of *community* diffusion is that node v can have the greatest global diffusion by accessing the important nodes with the highest degree. Moreover, if the node v has the minimum distance with the nodes with a maximum degree, it positively has better diffusion.

Table 1
Comparison of time complexity between different algorithms.

Algorithm	Time complexity
PHG	$O(nlogn + n + kn'm')$
CI	$O(n^2kR)$
VoteRank	$O(m + klogn + k\frac{m^2}{n^2}R)$
CTIM	$O(n' + (n' + m'logn'))$
K-core	$O(m + kn'R)$
LIR	$O(m + kn'R)$
ProbDeg	$O(klogn + m)$
TI-SC	$O(nlogn + m + Rn + (k-1)Rn')$
FIP algorithm	$O(nlogn + k'n'R)$

Algorithm 2 shows the computation of p_d for all nodes in C_b.

Algorithm 2: Calculate the probability coefficient of community diffusion for each node v in C_b

Input: $C_b = \{c_{b1}, c_{b2}, \cdots, c_{bx}\}$, $v_I = \{v_{I1}, v_{I2}, v_{I3}, \cdots, v_{Ih}\}$, $v_O = \{v_{O1}, v_{O2}, v_{O3}, \cdots, v_{Oh}\}$
Output: p_d for all nodes
1: **for each** c_{bi} in C_b **do**
2: **for each** v in c_{bi} **do**
3: **for each** v_{Ih} in v_I **do**
4: $p_d = p_{ar_i}(n_1 + 2\sqrt{n_2})$
5: **end for**
6: **for each** v_{Oh} in v_O
7: $p_d = \frac{|v_{Hc}|.deg_{c,max}}{\sum_{ci=1}^{n_c} deg_{(ci,max)}} \times \sum_{i=1}^{w} \frac{1}{(x_{sp_i} \times p_{ar_i})}$
8: **end for**
9: **Normalize** p_d of nodes
10: **end for**
21: **end for**
22: **return** p_d *of all nodes*

3.2.2. Generating candidate nodes

The candidate nodes include some nodes in a community that has better spread influence in comparison to other nodes with regard to the community topology. In fact, candidate nodes are the representative that has the potential to select a seed node. As a result, the selection of the candidate nodes has an important effect on the reduction of computational overhead and optimal selection of the seed nodes. The candidate nodes are divided into two main groups: the first group contains k overlapping nodes, and the second group is the best non-overlapping nodes. These two groups are selected in two steps as follow:

Step 1: selecting k important overlapping nodes:

Overlapping nodes are important because they are the communication path of several communities, so they disseminate much influence to other communities. For this reason, it is important to consider in selecting potential seed nodes. It should be noted that the number of overlapping nodes in large social networks is very high. For this reason, we cannot consider all overlapping nodes due to the high computational overhead. Besides, some overlapping has less effect on information diffusion due to their fewer connections to diverse communities. Therefore, to reduce this overhead time, overlapping nodes are ranked for communities with condition $w_{ci} \geq \theta_c$ using equation (6). In the end, the best overlapping nodes are added to set F. The set F is known as the candidate nodes.

$$n_{v_j} = cc_{v_j} - nb_{v_j} \tag{7}$$

where cc_{v_j} is the number of connections of the overlapping node with diverse communities and nb_{v_j} is the number of cc_{v_j} who has less degree than v_j. After computing n_{v_j} for each overlap node v_j, n_c nodes with the largest n_{v_j} are added to the set F, where n_c is the number of communities in C_b. Algorithm 3 presents the procedure for selecting the best overlapping nodes.

Algorithm 3: Select and add influential overlapping nodes to set F

Input: $C_b = \{c_{b1}, c_{b2}, \cdots, c_{bx}\}$
Output: F
1: initialize $F \rightarrow \varnothing$;
2: **for each** c_{bi} in C_b **do**
3: **for each** v_j in c_{bi} **do**
4: $n_{v_j} = cc_{v_j} - nb_{v_j}$
5: **end for**
6: **end for**
7: Rank the values of n_{v_j} in descending order
8: select the number of n_c nodes with the largest n_{v_j} and add to set F
9: **return** F // *F is the candidate set of overlapping nodes*

Step 2: Select the best non-overlapping influential nodes:

We continue the candidate selection from all nodes in set Cb in the second step, except the selected nodes in the previous step. In the selection of candidate and seed nodes, in addition to overlapping nodes, non-overlapping nodes are also important. Therefore, to generate a set of candidate nodes, the b_{inf,v_c} is computed for the node v_c by equation (8) as follow:

$$b_{inf,v_c} = \sum_{i=1}^{nn} deg_{v_i} + e^{(p_d + \sum_{i=1}^{g} w_c)} \sum_{i=1}^{ne} deg_{v_i} \tag{8}$$

where ne and nn respectively are the number of neighbors at a distance 1 and 2 from the node v_c, v_i is the neighbor node at a distance 1 and 2 from the node v_c, and deg_{v_i} is the degree of node v_i. Moreover, g represents the number of communities that node v_c has a connection with them. Nodes with high b_{inf,v_c} are powerful bridges for information diffusion between communities and may have high spread influence. Therefore, the nodes with high b_{inf,v_c} in each community c_i is selected and added to the set T_{c_i}. The size of the set T_{c_i} for each community c_i is defined by equation (9).

$$NT_{c_i} = w_{ci}\sqrt{k} \times \log(n_{ci}) \tag{9}$$

where NT_{c_i} is the length of the set T_{c_i}, n_{ci} is the number of nodes is the community c_i, and k is the size of the seed set.

After selecting the candidate sets by Algorithm 3 and 4, the final candidate nodes are defined by equation (10) as follow:

$$FC = F \cup T \tag{10}$$

Table 3
General properties of real-world and Synthetic networks that create with forest fire model.

Data set	M−Fo115	M−Fo120
Node	10 k	10 k
Edge	23 k	29 k
Max Degree	155	229
Min Degree	1	1
Number of communities	704	603

Table 2
General properties of the real-world.

Data set	DBLP	Email	Route views	Douban	Sister cities	PGP	As-22july06
Node	317 k	1 k	6 k	154 k	14 k	10 k	23 k
Edge	1 M	5 k	13 k	327 k	20 k	24 k	48 k
Max Degree	343	209	1459	287	99	205	2390
Min Degree	1	1	1	1	1	1	1
Number of communities	50,982	41	97	2599	472	734	308

Table 4
General properties of real-world and Synthetic networks created with LFR.

Data set	n	γ	β	μ	$< K >$	Max Degree	Min Community	Max Community
LFR1	1000	2	2	0.2	8	100	30	100
LFR2	1000	2	2	0.9	12	100	30	100
LFR3	2000	2	2	0.5	9	100	30	100
LFR4	5000	2	2	0.6	4	100	30	100

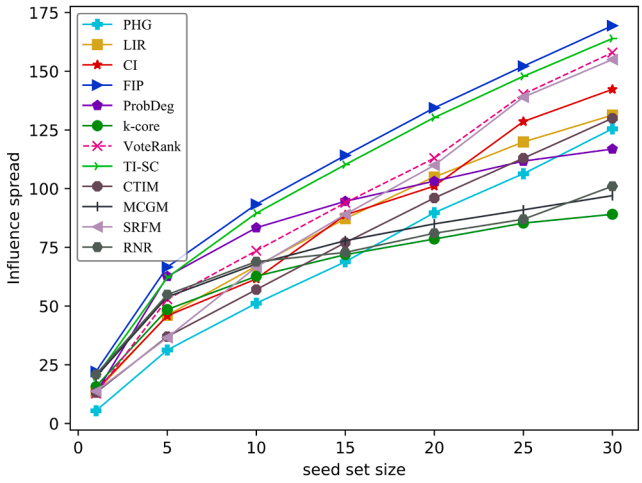


Fig. 4. Influence spreads of different algorithms on the DBLP. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

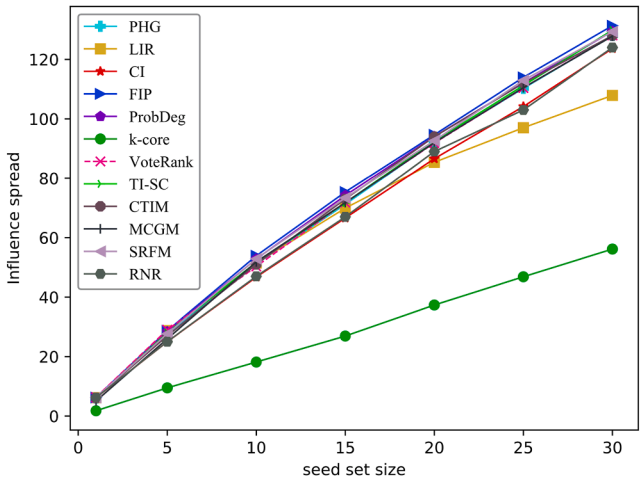


Fig. 5. Influence spreads of different algorithms on the Douban. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

Where *FC* is all candidate sets that are potentially considered for evaluating and selecting as seed nodes.

Algorithm 4: Select and add influential non-overlapping nodes to the candidate set T_{c_i}

Input: $C_b = \{c_{b1}, c_{b2}, \cdots, c_{bx}\}$
Output: F
1: **for each** c_{bi} in C_b **do**
2: **for each** v_c in c_{bi} **do**
3: $b_{inf,v_c} = \sum_{i=1}^{nn} deg_{v_i} + e^{(p_d + \sum_{i=1}^{s} w_c)} \sum_{i=1}^{ne} deg_{v_i}$
4: **end for**
5: **end for**
6: **for each** c_{bi} in C_b **do**
7: **Select the** NT_{c_i} nodes with the highest $b_{inf,v}$ for community c_{bi} and add to its T_{c_i} set
12: **end for**
13: **for each** c_{bi} in C_b **do**
14: $T \leftarrow \bigcup_{i=1}^{bi} T_{c_i}$ // *candidate set of non-overlapping nodes*
15: **end for**
17: **return T**

3.3. Selecting the final seed nodes

After generating a set of candidate nodes, the final seed nodes with the maximum influence spread are selected. All generated candidate nodes are evaluated, and the top k nodes with the best influence spread are added to the seed set. In other words, in the candidate set, the k node with the maximum influence spread under an independent cascade model and 1000 frequencies for the Monte Carlo simulation are considered as seed nodes. Therefore, the nodes within the set *FC* with the maximum influence spread are selected as the seed nodes using the function $f^*(S)$ (by equation (11)) in the independent cascade model.

$$f^*(S) = \operatorname{argmax} \sigma(S \cup \{v_H\}) \tag{11}$$

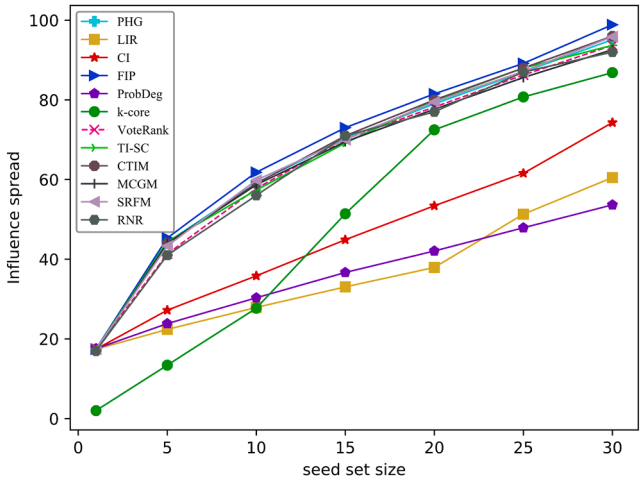


Fig. 6. Influence spreads of different algorithms on the Route views. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

In the function $f^*(S)$, S is the seed nodes, and $\sigma(S \cup \{v_H\})$ achieves the influence spread with regard to the addition of the node v_H (the nodes in the set *FC*) to the set S.

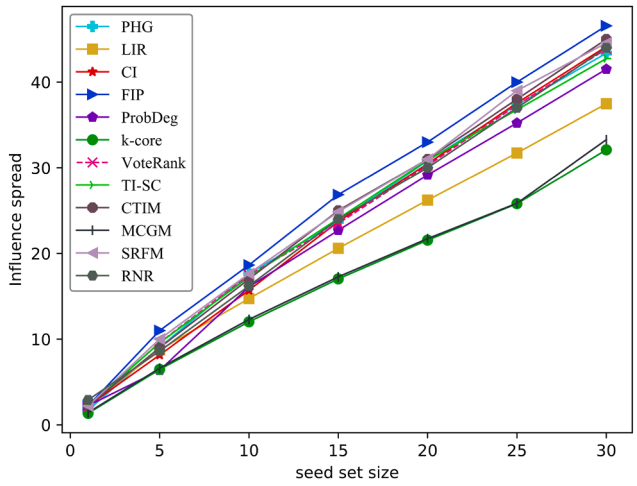


Fig. 7. Influence spreads of different algorithms on the Sister cities. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

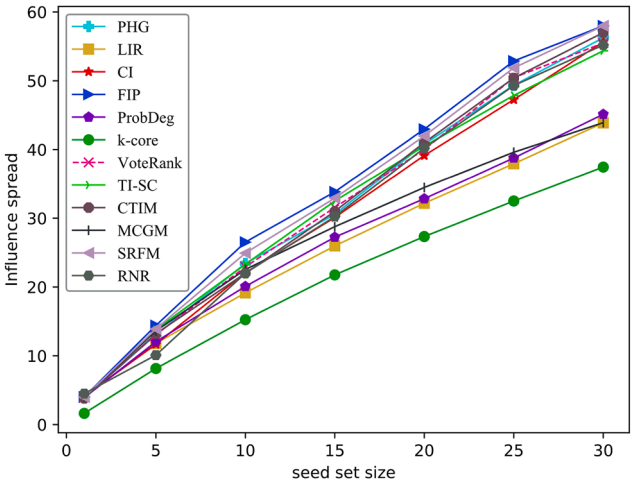


Fig. 9. Influence spreads of different algorithms on the PGP. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

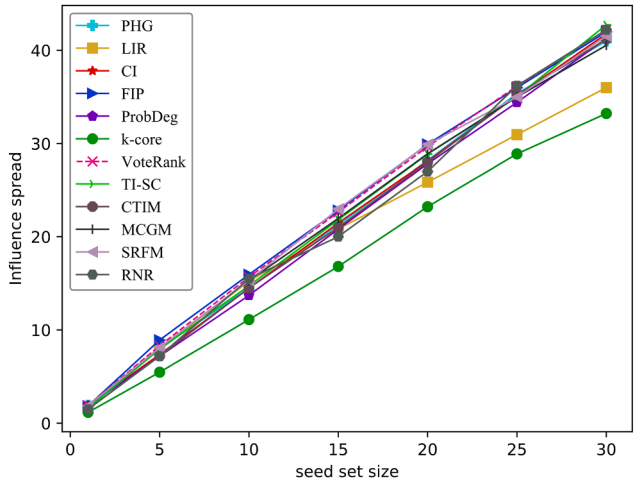


Fig. 8. Influence spreads of different algorithms on the Email. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

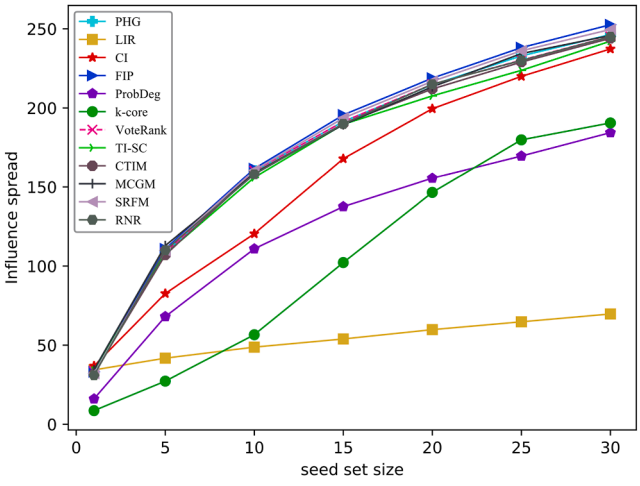


Fig. 10. Influence spreads of different algorithms on the As-22july06. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

3.4. Time complexity analysis

The time complexity analysis in the FIP algorithm includes two main steps. In the first step, the time complexity of community detection and reduction is $O(nlogn+n+c) \approx nlogn$. In the next step, the time complexity of the generating candidate set and selection of the final seed set is $O(kn'c' + k'n'R) \approx k'n'R$, Where c' is the number of communities in the set C_b. Moreover, n' is the nodes of all nodes in communities of set C_b and $n' \ll n$. Therefore, the total time complexity of the FIP algorithm is as $O(nlogn + k'n'R)$ where n is the number of the graph nodes and c stands for the number of the communities in a social network. k represents the number of seed nodes, k' is the number of candidate nodes (and $k' \ll kc'$) and R is the number of frequencies of the Monte Carlo simulation. The time complexity of our proposed algorithm and other basic and recently proposed algorithms are summarized in Table 1.

In the analysis of time complexity, $n' \ll n$, and $m' \ll m$. The LIR and K-core algorithms have less time complexity than other state-of-the-art algorithms. However, their performance is not satisfactory in terms of influence spread. The CI and PHG algorithms require high time complexity than the FIP algorithm. The time complexity and performance of the FIP algorithm in comparison to CI and PHG as two well-known methods are superior and more stable.

4. Experimental results and analysis

4.1. Dataset

In order to verify the proposed algorithm, both real-world and synthetic networks are used to evaluate the efficiency of the FIP algorithm

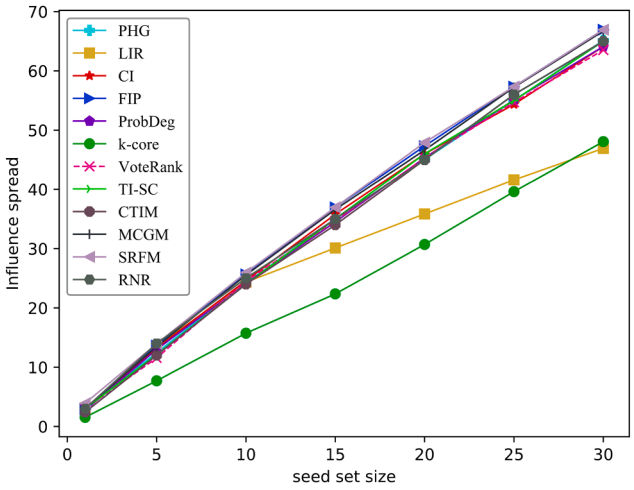


Fig. 11. Influence spreads of different algorithms on the M−FO115 network. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

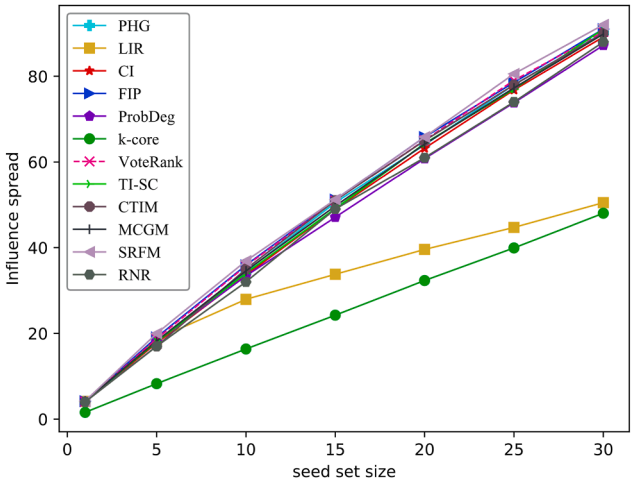


Fig. 12. Influence spreads of different algorithms on the M−FO120 network. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

Table 5
Parameters of LFR datasets.

Parameters	Description
n	Number of nodes
γ	Power law exponent for the degree distribution
β	Power law exponent for the community size distribution
μ	The fraction of intra-community edges incident to each node
$<K>$	the average degree of nodes

and other compared methods. The information on these networks is shown in Table 2, Table 3, and Table 4.

4.1.1. Real-world networks

We first evaluate the performance of algorithm FIP on seven real-world datasets. The datasets are undirected. Dataset's sizes are large

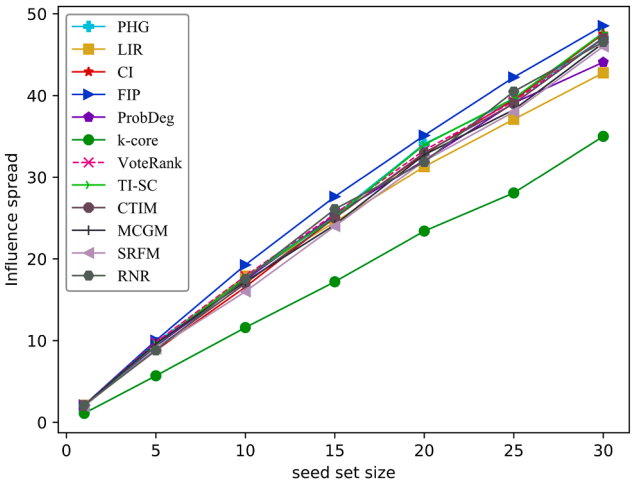


Fig. 13. Influence spreads of different algorithms on the LFR1. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

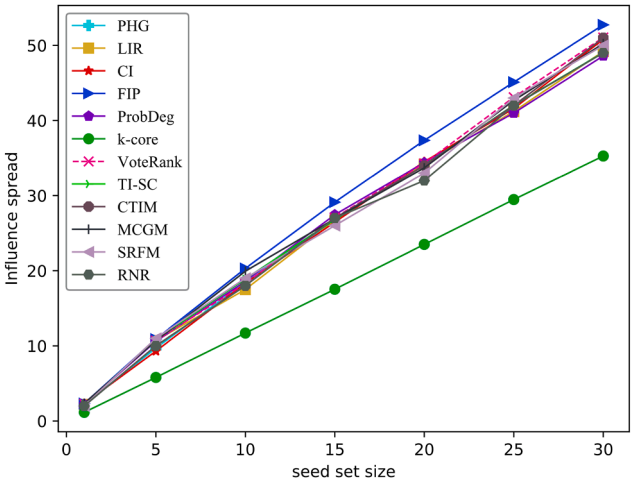


Fig. 14. Influence spreads of different algorithms on the LFR2. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

and medium. All networks are available on the KONECT[2] website.

- **DBLP:** The DBLP provides a co-authorship network in computer science (Leskovec, Kleinberg, & Faloutsos, 2007). If two authors have collaborated on at least one paper, an edge is created between them. The network consists of 317 K nodes and 1 M edges.
- **Email:** The email dataset is a network of the University at Rovira i Virgili (URV) that edges represent that at least one email was sent (Guimera, Danon, Diaz-Guilera, Giralt, & Arenas, 2003). The network contains 1 K nodes and 5 K edges.
- **Route views**: Dataset Route views are a network of autonomous systems that connect to each other via the Internet (Kunegis, 2013). The nodes represent autonomous systems, and the edges represent

[2] https://konect.cc/.

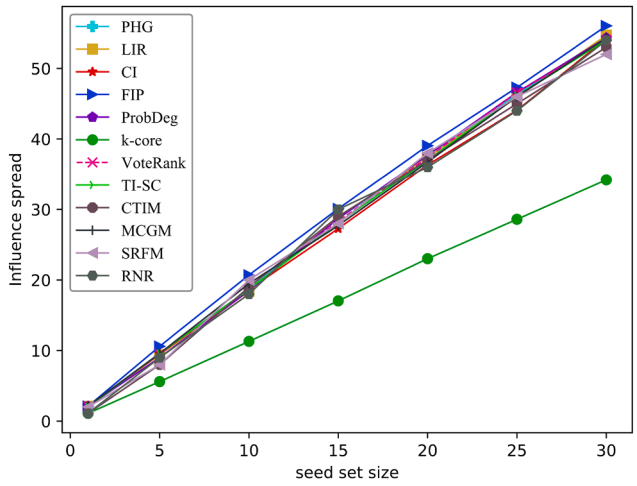


Fig. 15. Influence spreads of different algorithms on the LFR3. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

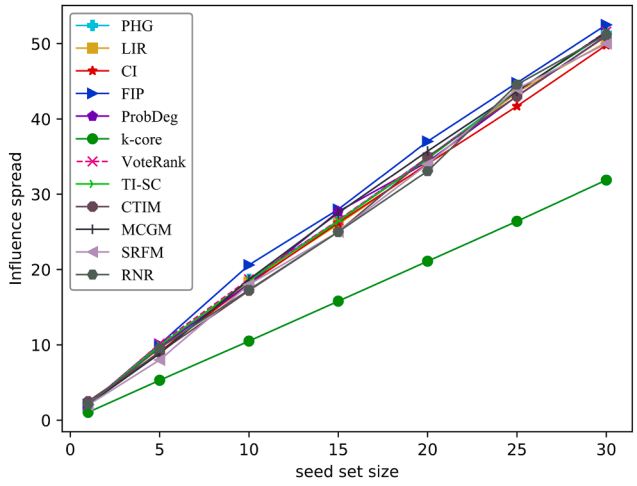


Fig. 16. Influence spreads of different algorithms on the LFR4. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

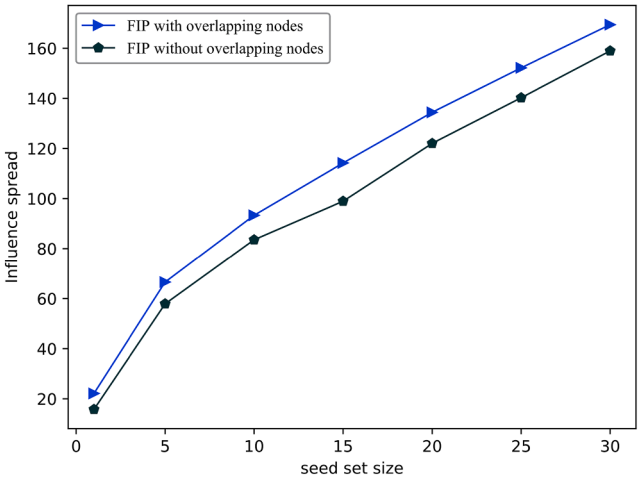


Fig. 17. Influence spreads of different algorithms on the DBLP. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

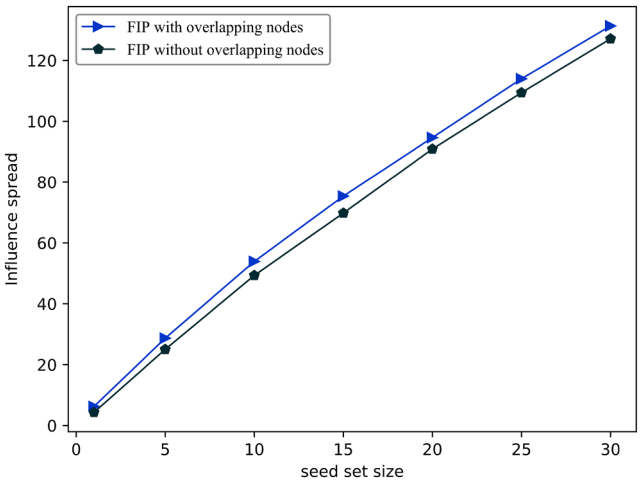


Fig. 18. Influence spreads of different algorithms on the Douban. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

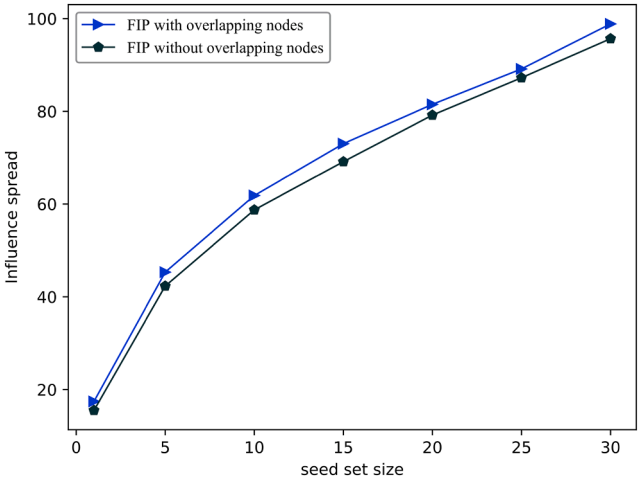


Fig. 19. Influence spreads of different algorithms on the Route views. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

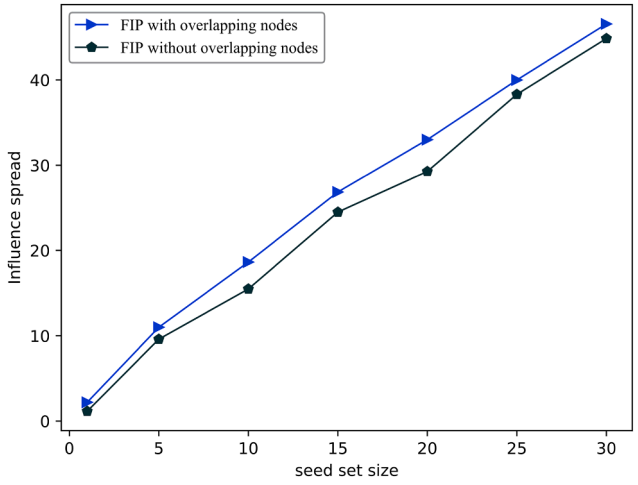


Fig. 20. Influence spreads of different algorithms on the Sister cities. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

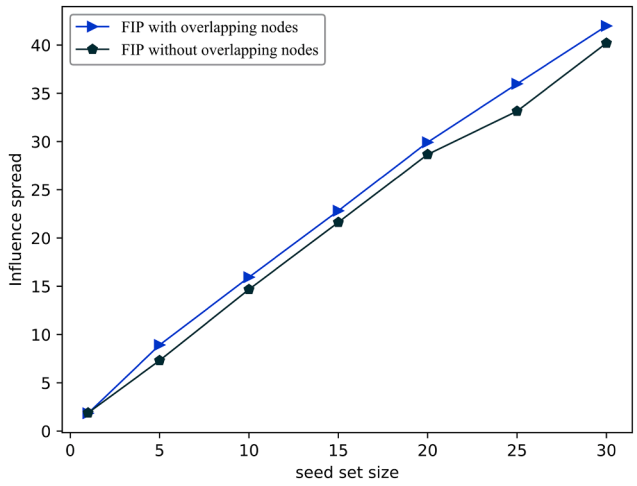


Fig. 21. Influence spreads of different algorithms on the Email. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

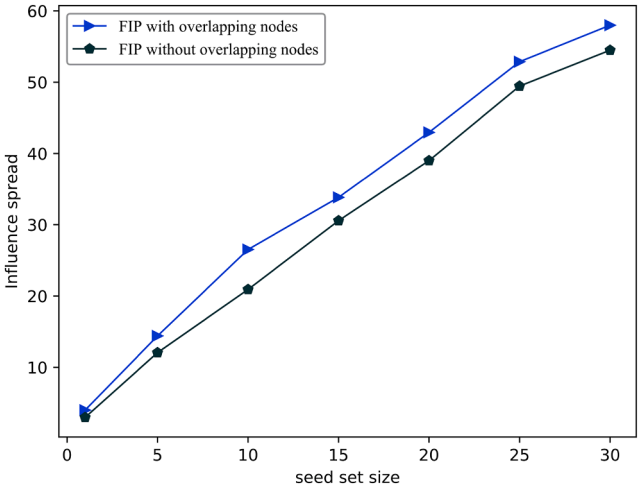


Fig. 22. Influence spreads of different algorithms on the PGP. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

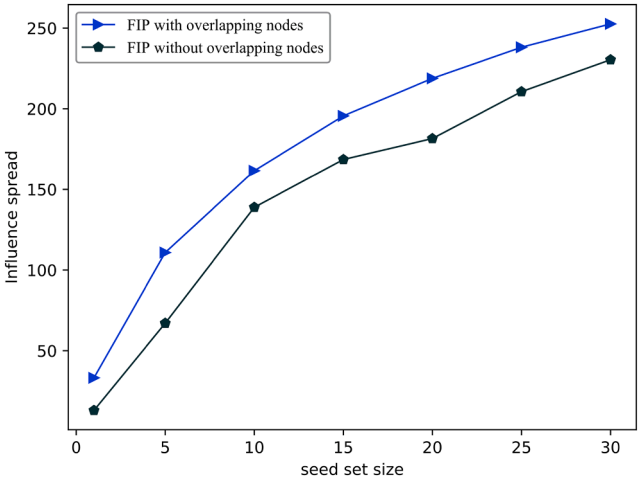


Fig. 23. Influence spreads of different algorithms on the As-22july06. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

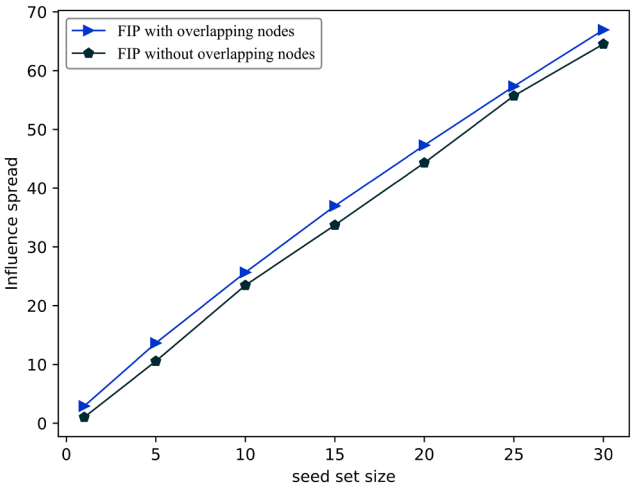


Fig. 24. Influence spreads of different algorithms on the M−FO115. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

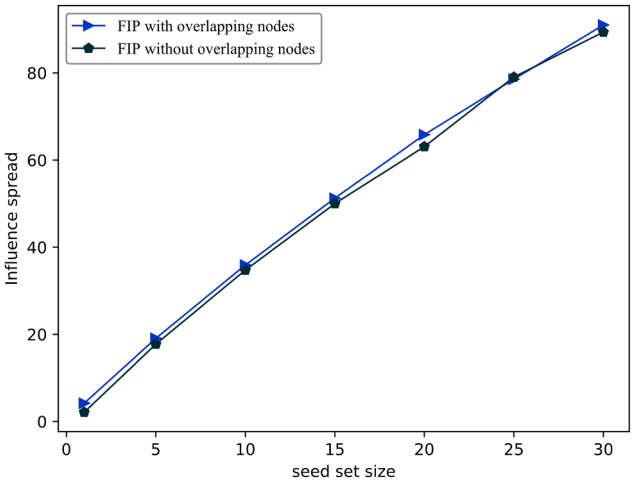


Fig. 25. Influence spreads of different algorithms on the M−FO120. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

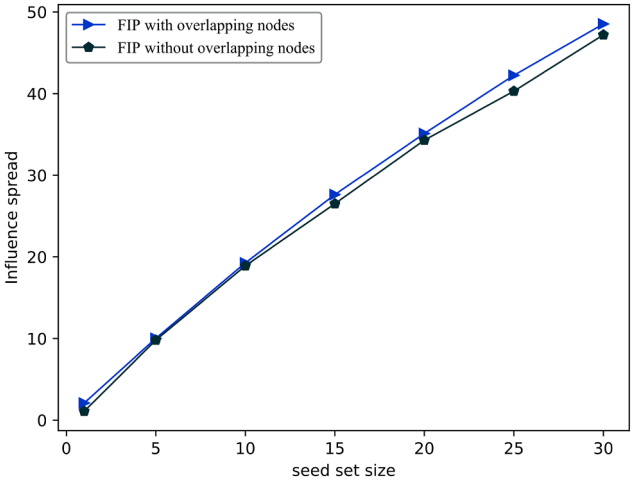


Fig. 26. Influence spreads of different algorithms on the LFR1. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

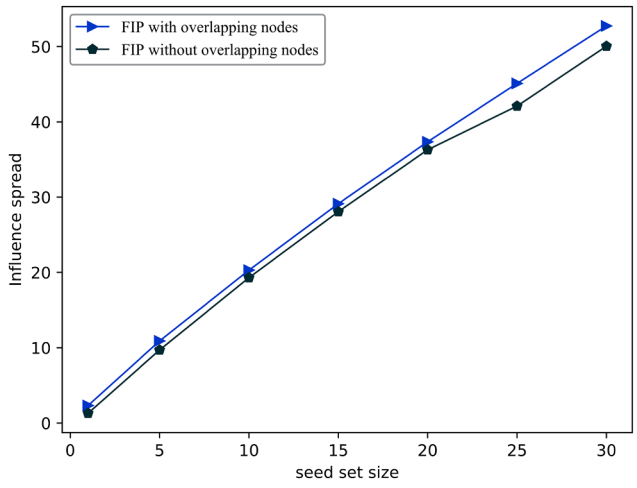


Fig. 27. Influence spreads of different algorithms on the LFR2. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

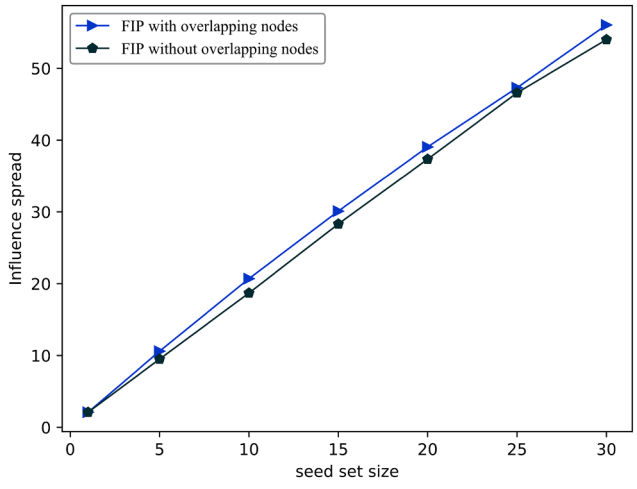


Fig. 28. Influence spreads of different algorithms on the LFR3. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

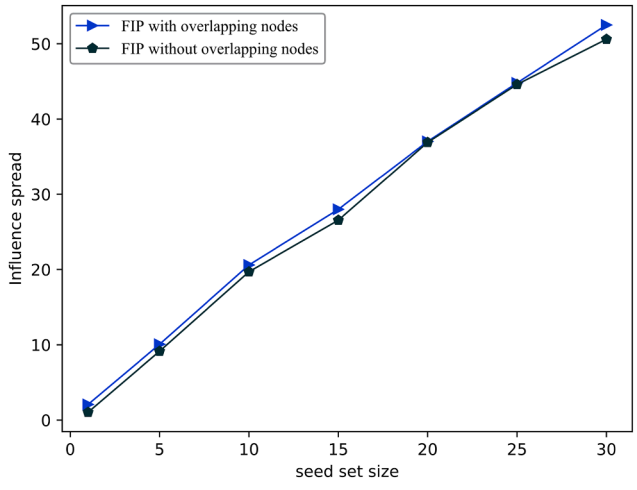


Fig. 29. Influence spreads of different algorithms on the LFR4. The x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread.

Table 6
Speedup % (in terms of influence spread) for FIP versus other methods on seven real-world datasets.

Data sets	Seed size	Algorithms (experiments based on the IC model)																					
		FIP	LIR	FIP	Kc	FIP	PHG	FIP	VR	FIP	CI	FIP	TI-SC	FIP	ProbDeg	FIP	CTIM	FIP	SRFM	FIP	MCGM
DBLP	10	37.9	−27.4	47.6	−32.2	80.8	−44.6	25.7	−20.4	50.2	−33.4	4.2	−4.08	11.9	−10.7	63.7	−38.9	40.1	−28.6	37.4	−27.2
	20	27	−21.2	69.9	−41.1	48.7	−32.7	18.0	−15.3	32	−24.2	3.1	−3.06	30.3	−23.2	40	−28.5	22.2	−18.1	58.1	−36.7
	30	28.9	−22.4	90.2	−47.4	34.9	−25.9	7.2	−6.7	19	−16.0	3.3	−3.2	44.9	−30.9	30.3	−23.2	9.3	−8.5	74.6	−42.7
Douban	10	5.6	−5.3	197.7	−66.4	3.4	−3.3	7.1	−6.6	15.4	−13.3	4.2	−4.05	2.03	−1.9	5.6	−5.3	1.8	−1.8	3.9	−3.7
	20	10.9	−9.8	153.6	−60.5	1.6	−1.5	3	−2.9	9.3	−8.5	2.26	−2.21	0.6	−0.6	0.6	−0.6	2.1	−2.1	2.8	−2.8
	30	21.7	−17.8	134	−57.2	1.7	−1.7	2.5	−2.5	6.2	−5.8	1.1	−1.14	1.4	−1.4	2.6	−2.5	1.5	−1.5	2.7	−2.6
Route views	10	115	−53.5	115.8	−53.6	1.1	−1.7	3.6	−3.5	67.3	−40.1	4.3	−4.15	97.3	−49.3	4.7	−4.5	3.3	−3.2	5.4	−5.1
	20	112	−52.9	11.1	−10.0	0.7	−0.7	3.0	−2.9	51.0	−33.7	0.88	−0.87	91.4	−47.7	1.8	−1.8	2.5	−2.4	4.9	−4.7
	30	7	−6.6	11.5	−10.3	1.7	−1.7	3.4	−3.3	30.2	−23.2	3.38	−3.27	80.6	−44.6	2.9	−2.8	3.1	−3.0	6.7	−6.3
Sister cities	10	19.7	−16.4	46.6	−31.8	0	0	1.1	−1.1	12.8	−11.3	2.19	−2.14	9.5	−8.7	9.6	−8.7	5.6	−5.3	51.6	−34.0
	20	21.7	−17.8	48.3	−32.6	0.6	−3.9	3.5	−5.3	5.6	−9.4	3.58	−3.46	9.7	−8.8	6.4	−6	6.4	−6.0	51.9	−34.1
	30	21.6	−17.8	42.1	−29.6	2.7	−4.83	5.08	−3.7	3.8	−7.6	6.6	−6.19	9.7	−8.9	3.4	−3.3	4.4	−4.2	39.9	−28.5
Email	10	4.6	−4.4	43.2	−30.1	10.4	−9.4	1.9	−1.8	10.4	−9.4	8.1	−7.56	16.2	−13.9	9.9	−9	4.1	−4.01	4.1	−4.0
	20	15.8	−13.7	28.8	−22.4	6.4	−6	0	0	6.4	−6	3.6	−3.5	7.5	−7.0	6.8	−6.4	0.06	0.06	3.5	−3.4
	30	16.7	−14.3	26.2	−20.7	2.6	−2.1	0	0	0.7	−0.7	−1.76	−1.79	1.7	−1.7	1.8	−1.8	1.06	−1.05	3.5	−3.4
PGP	10	28.2	−22	61.1	−37.9	4.7	−4.4	2.5	−2.4	11.3	−10.2	5.2	−4.98	22.12	−18.1	11.4	−10.2	6.4	−6.03	17.6	−14.9
	20	33.6	−25.1	57.1	−36.3	5.4	−5.1	4.8	−4.6	9.7	−8.8	6.0	−5.2	30.85	−23.5	4.7	−4.5	2.3	−2.3	24.6	−19.7
	30	31.2	−23.8	53.7	−34.9	0.5	−2.9	3.02	−0.1	3.4	−3.3	6.6	−6.2	28.4	−22.1	1.7	−1.7	0.0	0.0	32.0	−24.2
As-22july06	10	231	−69.8	185.9	−65.0	0.8	−0.8	0.7	−0.7	34.2	−25.4	3.5	−3.45	45.7	−31.3	1.6	−1.6	0.6	−0.6	1.9	−1.8
	20	265	−72.6	49.3	−33.0	1.9	−1.9	1.8	−1.7	9.6	−8.8	5.4	−5.12	40.6	−28.9	3.2	−3.1	0.9	−0.9	2.5	−2.5
	30	262	−72.4	32.5	−24.5	2.5	−2.4	3.0	−2.9	6.4	−6.06	4.2	−4.11	36.9	−27.0	3.5	−3.4	1.2	−1.1	2.7	−2.6

Table 7
Speedup % (in terms of influence spread) for FIP versus other methods on synthetic networks that create with Forest fire model.

Data sets	Seed size	Algorithms (experiments based on the IC model)																			
		FIP	LIR	FIP	Kc	FIP	PHG	FIP	VR	FIP	CI	FIP	TI-SC	FIP	ProbDeg	FIP	CTIM	FIP	SRFM	FIP	MCGM
M−Fo115	10	5.3	−5.0	63	−38.6	6.2	−5.8	4.9	−4.6	4.4	−4.2	6.85	−6.41	6.4	−6.06	6.8	−6.4	1.1	−1.1	0.4	−0.4
	20	32.1	−24.3	54	−35.3	5.1	−4.8	4.4	−4.2	3.0	−2.9	2.83	−2.75	4.3	−4.19	5.1	−4.8	1.2	−1.2	1.4	−1.4
	30	42.6	−29.8	39	−28.2	1.2	−1.1	6.02	−5.6	4.3	−4.1	2.97	−2.88	4.4	−4.2	2.9	−2.8	0.0	0.0	0.3	−0.3
M−Fo120	10	28.3	−22	119.6	−54.4	4.06	−3.9	0.2	−0.27	6.8	−6.42	5.5	−5.24	7.8	−6.9	2.5	−2.4	−2.7	2.7	3.8	−3.6
	20	66.5	−39.9	103.7	−50.9	2.81	−2.7	1.23	−1.21	4.1	−3.95	2.38	−2.32	8.29	−7.6	1.2	−1.2	0.0	0.0	2.6	−2.5
	30	80.3	−44.5	89.3	−47.1	0.88	−0.88	0.66	−0.66	2.0	−1.9	−0.02	0.02	7.3	−4.1	1.1	−1	−1.08	1.09	1.2	−1.2

Table 8
Speedup % (in terms of influence spread) for FIP versus other methods on synthetic networks that create with Forest fire model.

Data sets	Seed size	Algorithms (experiments based on the IC model)																			
		FIP	LIR	FIP	Kc	FIP	PHG	FIP	VR	FIP	CI	FIP	TI-SC	FIP	ProbDeg	FIP	CTIM	FIP	SRFM	FIP	MCGM
LFR1	10	7.5	−7.0	66.0	−39.7	8.7	−8.0	7.5	−7.06	19.3	−14.0	8.8	−8.0	10.6	−9.6	13.2	−11.7	20.3	−16.9	11.8	−10.5
	20	12.2	−10.8	50.0	−33.3	3.0	−2.9	5.4	−5.1	6.8	−6.4	3.3	−3.2	9.5	−8.6	6.4	−6	9.7	−8.8	7.2	−6.7
	30	13.4	−11.8	38.6	−27.8	3.1	−3.0	3.2	−3.1	2.06	−2.0	1.7	−1.7	10.1	−9.1	3.2	−3.1	5.5	−5.2	4.5	−4.3
LFR2	10	16.0	−13.7	73.5	−42.3	9.7	−8.8	10.9	−9.8	10.6	−9.6	7.4	−6.8	9.1	−8.3	12.7	−11.3	6.8	−6.4	1.8	−1.8
	20	9.2	−8.7	58.9	−37.0	9.2	−8.4	8.9	−8.1	8.2	−7.6	9.7	−8.8	8.6	−7.9	9.8	−8.9	13.2	−11.6	10.8	−9.7
	30	7.1	−6.7	49.4	−33.1	7.1	−6.7	3.3	−3.2	3.8	−3.6	7.5	−7.0	8.5	−7.8	3.4	−3.2	5.4	−5.1	4.9	−4.7
LFR3	10	12.5	−11.1	83.1	−45.4	10.6	−9.6	10.6	−9.6	10.9	−9.8	10.6	−9.6	11.8	−10.6	8.9	−8.2	3.4	−3.3	6.2	−5.8
	20	4.4	−4.2	69.6	−41.0	4.4	−4.2	4.5	−4.3	7.5	−6.9	5.5	−5.2	3.6	−3.4	5.5	−5.2	2.7	−2.7	6.3	−5.9
	30	2.4	−2.3	63.8	−38.1	2.7	−2.6	3.7	−3.6	3.3	−3.2	3.6	−3.5	3.1	−3.0	5.7	−5.4	7.7	−7.2	3.2	−3.1
LFR4	10	12.5	−11.9	96.1	−49.0	10.1	−9.2	10.1	−9.2	13.8	−12.1	11.3	−10.1	13.8	−12.1	19.7	−16.5	14.4	−12.6	10.3	−9.3
	20	8.6	−7.9	75.4	−43.0	6.2	−5.7	6.1	−5.7	8.7	−8.0	6.7	−6.2	7.9	−7.3	5.8	−5.4	8.9	−8.1	3.6	−3.5
	30	4.6	−4.4	64.6	−39.2	2.1	−2.0	1.7	−1.7	5.4	−5.1	2.2	−2.1	2.8	−2.1	2.9	−2.8	5.0	−4.7	1.8	−1.8

Table 9
The average number of influences spread in k = 1 to k = 30 on seven real-world datasets.

Algorithms	DBLP	Douban	Route views	Sister cities	Email	PGP	As-22july06
PHG	88.7	91.45	78.3	31.2	27.8	40.3	207.06
CI	101.6	85.6	54.4	28.8	28.06	38.9	185.7
FIP	131.7	93.6	79.03	31.6	29.2	41.6	211.0
K-core	76.72	37.1	62.3	21.86	22.5	26.3	131.2
VoteRank	114.81	90.06	76.4	30.4	29.1	40.7	206.8
LIR	101.08	81.5	42.03	26.1	25.6	31.6	59.4
TI-SC	126.9	91.37	76.9	30.2	28.7	39.3	201.9
ProbDeg	101.1	91.8	42.0	28.9	27.5	32.6	150.2
CTIM	94.3	91.0	78.3	31.0	27.9	40.8	205.0
SRFM	110.5	91.6	78.4	31.06	28.9	41.6	209.00
MCGM	83.2	90.5	76.3	22.4	28.4	33.6	205.9
RNR	83.26	86.6	75.0	30.06	28.23	39.13	206.0

Table 10
The average of influence spread in k = 1 to k = 30 on synthetic networks that create with Forest fire model.

		M−FO115 dataset	M−FO120 dataset
Algorithms	PHG	45.03	62.8
	CI	44.8	61.9
	FIP	46.6	64.1
	K-core	31.4	32.3
	VoteRank	44.4	63.3
	LIR	35.6	39.2
	TI-SC	45.01	63.1
	ProbDeg	44.51	60.4
	CTIM	44.6	63.4
	SRFM	46.9	64.9
	MCGM	46.2	92.8
	RNR	45.0	60.33

Table 11
The average of influence spread in k = 1 to k = 30 on synthetic networks that create with LFR.

Algorithms	LFR1	LFR2	LFR3	LFR4
PHG	32.9	33.9	36.8	35
CI	32.3	34.5	36.3	33.98
FIP	34.3	36.8	38.6	36.71
K-core	23.3	23.4	22.8	21.16
VoteRank	32.7	34.55	36.6	35.06
LIR	30.66	33.63	36.8	34.19
TI-SC	33.12	33.9	36.5	34.85
ProbDeg	31.1	33.86	36.8	34.48
CTIM	32.6	36.7	36.7	34.4
SRFM	31.3	34.0	36.6	34.0
MCGM	32.1	34.6	36.8	35.3
RNR	31.99	33.0	36.0	33.85

the relationship between them. The network consists of 6 k nodes and 13 k edges.

- **Douban**[1]**:** The Douban is a network of Chinese social networking service. The network consists of 154 k nodes and 327 k edges.
- **Sister cities:** The Sister cities dataset is a network of cities of the world connected by "sister city" that extracted from the WikiData (Kunegis, 2013). The network consists of 14 k nodes and 20 k edges.
- **PGP:** The PGP dataset is the interaction network of users of the Pretty Good Privacy algorithm (Boguná, Pastor-Satorras, Díaz-Guilera, & Arenas, 2004). The network contains only the giant connected component. The network consists of 10 k nodes and 24 k edges.
- **As-22july06:** This is a dataset of the structure of the Internet at the level of autonomous systems. The network consists of 23 k nodes and 48 k edges (Kunegis, 2013).

4.1.2. Synthetic networks

We evaluate the FIP algorithm's performance through synthetic networks, and the forest fire model (Barabási & Albert, 1999) and LFR algorithm (Lancichinetti, Fortunato, & Radicchi, 2008) are used to create Synthetic networks.

- Forest fire model: A forward burning probability (p) is one parameter in this model. v first chooses a node w uniformly at random and forms a link to w. Next, a random number x is generated that is binomially distributed with a mean $(1-p)^{-1}$. Node v selects x edges with a node with node w. This model has the power-law distribution property, which makes the Synthetic network closer to the real world. Synthetic networks are as follows:

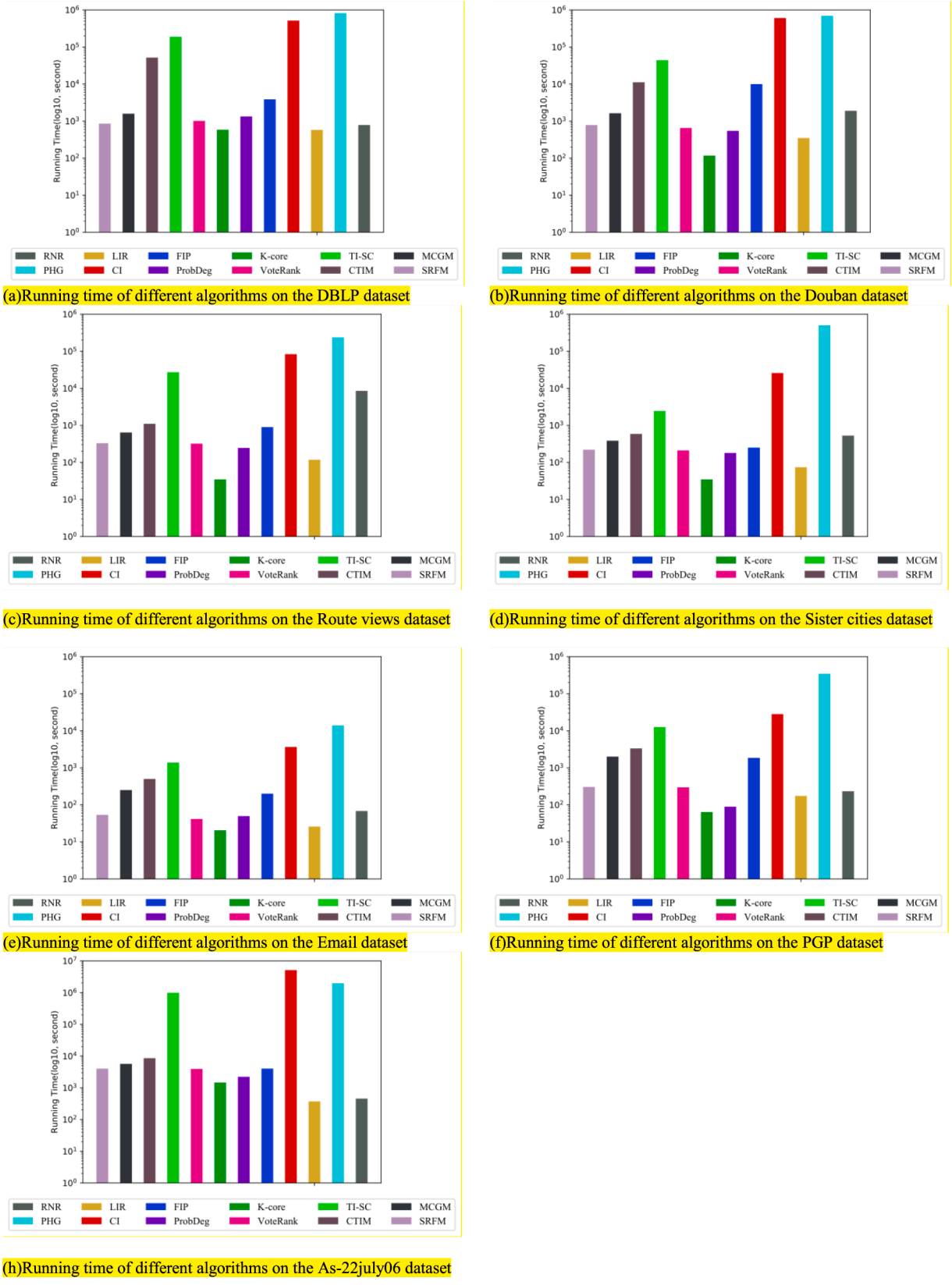
- **M−Fo115:** The M−Fo115 network generated from the forest fire model with connecting probability $p = 0.115$, which includes 10 k nodes and 23 k.
- **M−Fo120:** The M−Fo120 network generated from the forest fire model with connecting probability p = 0.120, which includes 10 k nodes and 29 k.

- LFR synthetic networks: we choose the LFR algorithm to generate synthetic networks. In this algorithm, both the degree and the community size distributions are power laws with different exponents.

4.2. Basic compared algorithms

The FIP algorithm has been compared with eight basic algorithms. The list of basic algorithms is described below.

- **PHG:** Qiu et al. developed a PHG algorithm that uses a community-based approach. It also uses the greedy algorithm to find influence spread (Qiu et al., 2019).
- **TI-SC:** Ahmadi Beni et al. developed a community-based algorithm to examine the relationships between the core nodes and the scoring ability of other nodes in communities [37].
- **MCGM:** Li et al. proposed a new gravity model to identify influential spreaders based on different features like k-shell value and eigenvector centrality value (Z. Li & Huang, 2022).
- **ProbDeg:** Nguyen et al. developed the ProbDeg that uses multi-hop neighbors and propagation probabilities of nodes to select seed nodes [33].



(a)Running time of different algorithms on the DBLP dataset

(b)Running time of different algorithms on the Douban dataset

(c)Running time of different algorithms on the Route views dataset

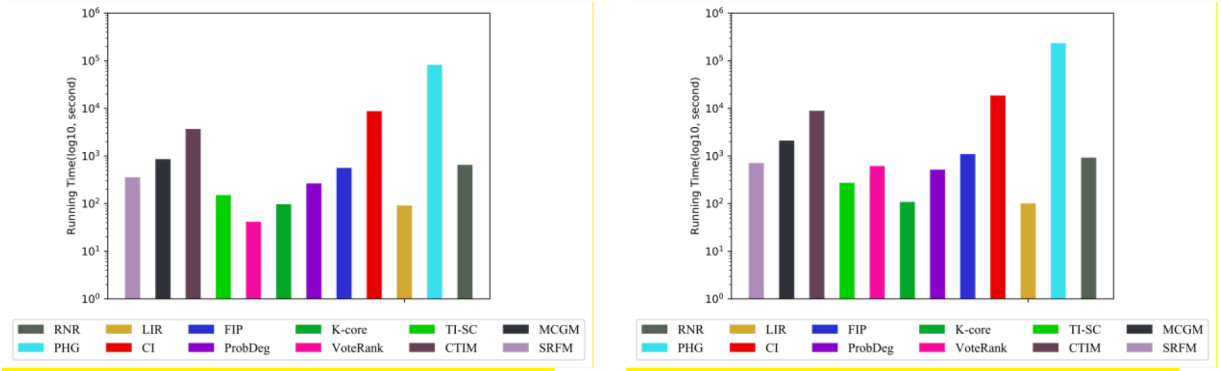
(d)Running time of different algorithms on the Sister cities dataset

(e)Running time of different algorithms on the Email dataset

(f)Running time of different algorithms on the PGP dataset

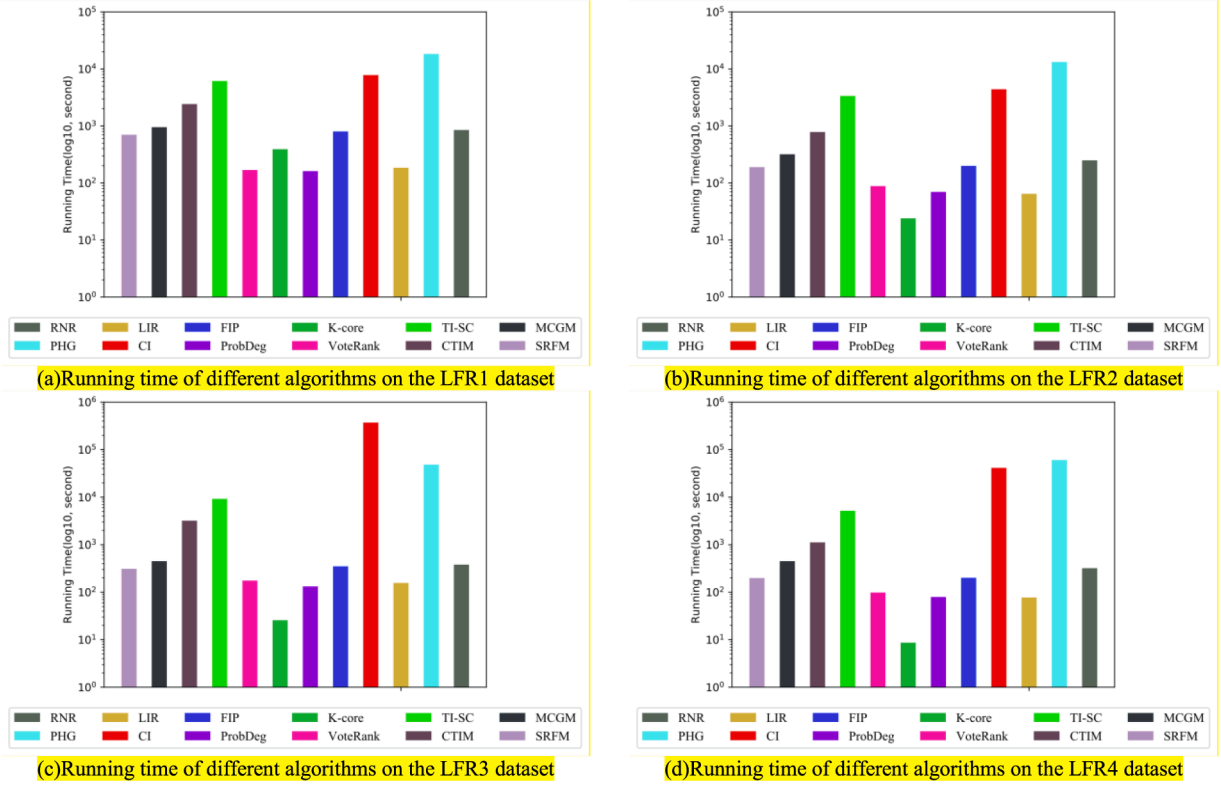
(h)Running time of different algorithms on the As-22july06 dataset

Fig. 30. Running time of different algorithms on seven real-world datasets (k = 30).



(a)Running time of different algorithms on the M-FO115 dataset (b)Running time of different algorithms on the M-FO120 dataset

Fig. 31. Running time of different algorithms on synthetic networks (k = 30) that create with forest fire model.



(a)Running time of different algorithms on the LFR1 dataset (b)Running time of different algorithms on the LFR2 dataset

(c)Running time of different algorithms on the LFR3 dataset (d)Running time of different algorithms on the LFR4 dataset

Fig. 32. Running time of different algorithms on synthetic networks (k = 30) that create with LFR.

- **LIR:** Liu et al. developed the LIR algorithm based on heuristic methods (Liu et al., 2017). In the LIR algorithm, the LI value for each node is computed according to the degree of neighbors, and then, the set of nodes with the lowest LI value is sorted in descending order, and k nodes are selected as seeds.
- **SRFM:** Ahmadi Beni et al. proposed the SRFM algorithm based on core nodes (Ahmadi Beni & Bouyer, 2021).
- **Collective Influence(CI):** Morone et al. proposed the CI algorithm based on localization of influence spread computation (Morone et al., 2016). In this algorithm, the influence spread computation in a circle is limited to the radius L.
- **VoteRank:** Zhang et al. developed the VoteRank algorithm based on voting (J.-X. Zhang et al., 2016).
- **K-core:** Kitsak et al. Developed a K-core algorithm that specifies the core and periphery nodes in the graph and considers the core nodes as seed nodes (Kitsak et al., 2010).
- **CTIM:** Kazemzadeh et al. proposed the CTIM algorithm that there is a positive correlation between influential nodes and high charismatic power in this algorithm (Kazemzadeh et al., 2022).
- **RNR:** Xiaobin et al. proposed a new algorithm for selecting influential nodes by reversed rank, weights, and influence power (Rui et al., 2019).

4.3. Evaluation metrics

We use three metrics to evaluate the efficiency of the FIP algorithm:

- **The influence spread:** This metric measures the accuracy of seed nodes in information diffusion. The higher value for this metric reveals that the seed nodes have properly been selected for the influence maximization problem. Thus, the influence spread is the average number of activated nodes for each iteration of the Monte Carlo simulation in the independent cascade model.

- **Runtime:** The runtime is measured in FIP, and all compared algorithms to select k = 30 seed nodes.
- **Speedup:** The speedup is measured for FIP over baseline algorithms with k = 10, 20, and 30 seed nodes.

4.4. Experiment setup

In this paper, The FIP algorithm is programmed in Python language and implemented on a computer with 2.5 GHz Intel Core i5 CPU-3230 M and 12 GB memory. We tested our proposed algorithm on thirteen networks and evaluated them with seven state-of-the-art algorithms. The diffusion model and influence probability generation are important issues in influence maximization. The FIP algorithm uses the independent cascade (IC) model. The influence probability from node u to v is $p_{uv} = 0.01$.

4.5. Result

4.5.1. Influence spread

We first compare the influence spread of different algorithms on seven real-world datasets where the x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread. The results in four real-world datasets represent that the FIP algorithm completely outperforms other compared methods in terms of influence spread. The K-core algorithm shows the worst performance on all networks except the Route views and As-22july06 datasets. The PHG method, though performs well on some datasets, cannot provide any performance guarantee. For example, Figs. 4 and 5 show that their influence spread values are weak in datasets DBLP and Douban. However, Fig. 56 represents that FIP, PHG, CTIM, and TI-SC algorithms exhibit the same influence spread value. When k is 15, in Fig. 6, the influence spread values of the PHG and VoteRank algorithms are the same with the FIP algorithm. Fig. 7 also shows that the FIP algorithm has better performance than others. Fig. 8 represents that FIP and VoteRank algorithms exhibit the same influence spread value. We can observe significant gaps of influence spread value between the FIP algorithm and other algorithms in Fig. 9. Overall, from the results on the real-world networks, it is concluded that the FIP algorithm has better efficiency than state-of-the-art algorithms in finding top influential seed nodes. Furthermore, the PHG method has the second-best performance. In Fig. 10, we see that the influence spread values of PHG, VoteRank, and TI-SC algorithms are lower than the FIP algorithm. For example, on the As-22july06 dataset with k = 30, the FIP algorithm achieves an influence spread value of 252.599, while the value of VoteRank and PHG are 245.142 and 246.358, respectively.

On the other hand, two synthetic networks are used to compare the influence spread. It is obvious in Fig. 11 that the FIP algorithm outperforms other compared algorithms in the M−FO115 dataset. Additionally, Fig. 12 shows that FIP, VoteRank, PHG, and TI-SC algorithms exhibit the same influence spread value in the M−FO120 dataset. We also analyze the average influence spread in k = 10, k = 20, and k = 30 on synthetic networks in Table 5. We can observe that FIP obtains the optimal influence spread among all algorithms. Fig. 13 also shows that the FIP algorithm has better performance than others. In Fig. 14 and Fig. 15, and Fig. 16 we see that the influence spread values of CI, PHG, VoteRank, and TI-SC algorithms are lower than the FIP algorithm.

Now, we compare the influence spread of the FIP algorithm with overlapping nodes and the FIP algorithm without overlapping nodes on seven real-world datasets, six synthetic networks where the x-axis represents the number of seed nodes. In contrast, the y-axis represents the overall influence spread. The results in all real-world datasets represent that the FIP algorithm with overlapping nodes outperforms in terms of influence spread. In FIP algorithm, overlapping nodes have a significant effect on propagation. If these nodes do not affect the selection of seed nodes, the spread of influence will be reduced dramatically. In Figs. 17-29, it can be seen that the influence spread without overlapping nodes has decreased in different datasets.

Table 6 shows the speedup % (in terms of influence spread) of our proposed algorithm FIP over baseline algorithms. The speedup is computed using equation (12).

$$Speedup = ((A - B)/A) \times 100 \tag{12}$$

For example, if the influence rate for the FIP and LIR algorithms are 162.42 and 131.336, respectively, the speed of the FIP compare to LIR is calculated as follows:

$$speed - up_{IOG \to LIR} = ((162.42 - 131.336) \div 169.42) \times 100 = 18.3$$

Also, the speed of the LIR compare to FIP is calculated as follows:

$$speed - up_{LIR \to IOG} = ((131.336 - 162.42) \div 131.336) \times 100 = -23.66$$

Table 6, Table 7, and Table 8 reveal that the FIP algorithm has positive speedup over baseline approaches in all networks. In these tables, the K-core algorithm called Kc, and the VoteRank algorithm called VR. We also analyze the average influence spread in k = 1 to k = 30 in Table 9, Table 10, and Table 11. We can observe that FIP obtains the optimal influence spread among all algorithms.

4.5.2. Runtime metric

Fig. 30, Fig. 31 and Fig. 32 show the runtime of different algorithms on the seven real-worlds, two synthetic networks that create with forest fire model, and four synthetic networks that create with LFR, respectively. Here the running time is the time of selecting k = 30 seed nodes. From the results, we see that the running time of LIR, k-core, and VoteRank is low on all datasets, but all of them have the lowest quality compared to other methods; and they cannot provide any performance guarantee in terms of influence spread. Nevertheless, the FIP has the best run time in comparison to PHG and CI algorithms. The worst runtime is for the PHG algorithm that is a community-based approach like the FIP algorithm.

5. Conclusions

In this paper, we proposed a new community-based algorithm for focusing on the effect of overlapping nodes and the probability coefficient of community diffusion theory to solve the influence maximization problem. We focused on two issues in this problem, the first effectiveness of seed and the second time complexity. Thus, the FIP algorithm considers two main steps to improve the effectiveness of selected seed nodes: 1. Generation and optimization of initial communities, and 2. Generating the candidate nodes and selecting the final seed set. To reduce the search space in seed nodes selection, communities that do not have required spread influence are not considered in computations of candidate nodes selection. In the FIP algorithm, social relationships and community structure are analyzed to improve seed node selection performance. We verified the quality and running time measures of FIP on seven real-world datasets and 2 artificial networks. Experimental results demonstrated the effectiveness and runtime efficiency of the FIP algorithm. Results proved that our algorithm could significantly outperform the baseline algorithms in terms of time efficiency with no compromise on time complexity. As future work, the FIP algorithm can be extended under the Linear Threshold Model and consider the effect of distance among candidate nodes.

CRediT authorship contribution statement

Asgarali Bouyer: Conceptualization, Methodology, Software, Writing – review & editing, Supervision. **Hamid Ahmadi Beni:** Methodology. **Bahman Arasteh:** Conceptualization, Software, Writing – review & editing. **Zahra Aghaee:** Writing – review & editing. **Reza Ghanbarzadeh:** Methodology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References:

Aghaalizadeh, S., Afshord, S. T., Bouyer, A., & Anari, B. (2021). A three-stage algorithm for local community detection based on the high node importance ranking in social networks. *Physica A: Statistical Mechanics and its Applications, 563*, Article 125420.

Aghaee, Z., Beni, H. A., Kianian, S., & Vahidipour, M. (2020). A heuristic algorithm focusing on the rich-club phenomenon for the influence maximization problem in social networks. *Paper presented at the 2020 6th International Conference on Web Research (ICWR)*.

Aghaee, Z., Ghasemi, M. M., Beni, H. A., Bouyer, A., & Fatemi, A. (2021). A survey on meta-heuristic algorithms for the influence maximization problem in the social networks. *Computing, 103*(11), 2437–2477.

Ahajjam, S., & Badir, H. (2018). Identification of influential spreaders in complex networks using HybridRank algorithm. *Scientific Reports, 8*(1), 11932.

Ahmadi Beni, H., & Bouyer, A. (2021). Identifying influential nodes using a shell-based ranking and filtering method in social networks. *Big Data, 9*(3), 219–232.

Backstrom, L., & Kleinberg, J. (2014). Romantic partnerships and the dispersion of social ties: A network analysis of relationship status on facebook. *Paper presented at the Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*.

Banerjee, S., Jenamani, M., & Pratihar, D. K. (2019). ComBIM: A community-based solution approach for the Budgeted Influence Maximization Problem. *Expert Systems with Applications, 125*, 1–13.

Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *science, 286*(5439), 509–512.

Beni, H. A., & Bouyer, A. (2020). TI-SC: Top-k influential nodes selection based on community detection and scoring criteria in social networks. *Journal of Ambient Intelligence and Humanized Computing*, 1–20.

Berahmand, K., Bouyer, A., & Samadi, N. (2018). A new centrality measure based on the negative and positive effects of clustering coefficient for identifying influential spreaders in complex networks. *Chaos, Solitons & Fractals, 110*, 41–54.

Boguná, M., Pastor-Satorras, R., Díaz-Guilera, A., & Arenas, A. (2004). Models of social networks based on social distance attachment. *Physical review E, 70*(5), Article 056122.

Bouyer, A., Azad, K., & Rouhi, A. (2022). A fast community detection algorithm using a local and multi-level label diffusion method in social networks. *International Journal of General Systems*, 1–34.

Bouyer, A., & Beni, H. A. (2022). Influence maximization problem by leveraging the local traveling and node labeling method for discovering most influential nodes in social networks. *Physica A: Statistical Mechanics and its Applications*, 126841.

Bouyer, A., & Roghani, H. (2020). LSMD: A fast and robust local community detection starting from low degree nodes in social networks. *Future Generation Computer Systems, 113*, 41–57.

Chaharborj, S. S., Nabi, K. N., Feng, K. L., Chaharborj, S. S., & Phang, P. S. (2022). Controlling COVID-19 transmission with isolation of influential nodes. *Chaos, Solitons & Fractals, 159*, Article 112035.

Chakrapani, H. B., Chourasia, S., Gupta, S., & Haldar, R. (2021). Effective utilisation of influence maximization technique for the identification of significant nodes in breast cancer gene networks. *Computers in Biology and Medicine, 133*, Article 104378.

Chen, W., Wang, Y., & Yang, S. (2009). Efficient influence maximization in social networks. *Paper presented at the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Chen, Y.-C., Zhu, W.-Y., Peng, W.-C., Lee, W.-C., & Lee, S.-Y. (2014). CIM: Community-based influence maximization in social networks. *ACM Transactions on Intelligent Systems and Technology (TIST), 5*(2), 25.

Cheng, S., Shen, H., Huang, J., Zhang, G., & Cheng, X. (2013). Staticgreedy: Solving the scalability-accuracy dilemma in influence maximization. *Paper presented at the Proceedings of the 22nd ACM international conference on Information & Knowledge Management*.

Cherifi, H., Palla, G., Szymanski, B. K., & Lu, X. (2019). On community structure in complex networks: Challenges and opportunities. *Applied Network Science, 4*(1), 1–35.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. *Paper presented at the Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*.

Feng, C., Fu, L., Jiang, B., Zhang, H., Wang, X., Tang, F., & Chen, G. (2022). Neighborhood matters: Influence maximization in social networks with limited access. *IEEE Transactions on Knowledge and Data Engineering, 34*(6), 2844–2859. https://doi.org/10.1109/TKDE.2020.3015387

Ghalmane, Z., Cherifi, C., Cherifi, H., & El Hassouni, M. (2019). Centrality in complex networks with overlapping community structure. *Scientific reports, 9*(1), 1–29.

Ghalmane, Z., El Hassouni, M., Cherifi, C., & Cherifi, H. (2019). Centrality in modular networks. *EPJ Data Science, 8*(1), 15.

Gong, K., Tang, M., Hui, P. M., Zhang, H. F., Younghae, D., & Lai, Y.-C. (2013). An efficient immunization strategy for community networks. *PloS One, 8*(12).

Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., & Arenas, A. (2003). Self-similar community structure in a network of human interactions. *Physical Review E, 68*(6), Article 065103.

Kazemzadeh, F., Karian, A., Safaei, A. A., & Mirzarezaee, M. (2021). Intelligent Filtering of Graph Shells in the Problem of Influence Maximization Based on the Independent Cascade Model. *Paper presented at the 2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*.

Kazemzadeh, F., Safaei, A. A., & Mirzarezaee, M. (2022). *Influence maximization in social networks using effective community detection* (p. 127314). Physica A: Statistical Mechanics and its Applications.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. *Paper presented at the Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., & Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature physics, 6*(11), 888.

Kumar, S., Gupta, A., & Khatri, I. (2022). CSR: A community based spreaders ranking algorithm for influence maximization in social networks. *World Wide Web*, 1–20.

Kunegis, J. (2013). Konect: The koblenz network collection. *Paper presented at the Proceedings of the 22nd International Conference on World Wide Web*.

Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical review E, 78*(4), Article 046110.

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2007). Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD), 1*(1), 2.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. *Paper presented at the Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Li, W., Fan, Y., Mo, J., Liu, W., Wang, C., Xin, M., & Jin, Q. (2020). Three-hop velocity attenuation propagation model for influence maximization in social networks. *World Wide Web, 23*(2), 1261–1273.

Li, W., Li, Y., Liu, W., & Wang, C. (2022). An influence maximization method based on crowd emotion under an emotion-based attribute social network. *Information Processing & Management, 59*(2), Article 102818.

Li, W., Li, Z., Luvembe, A. M., & Yang, C. (2021). Influence maximization algorithm based on Gaussian propagation model. *Information Sciences, 568*, 386–402.

Li, W., Zhong, K., Wang, J., & Chen, D. (2021). A dynamic algorithm based on cohesive entropy for influence maximization in social networks. *Expert Systems with Applications, 169*, Article 114207.

Li, Z., & Huang, X. (2022). Identifying influential spreaders by gravity model considering multi-characteristics of nodes. *Scientific Reports, 12*(1), 1–11.

Lin, Y., Zhang, X., Xia, L., Ren, Y., & Li, W. (2019). A hybrid algorithm for influence maximization of social networks. *Paper presented at the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*.

Liu, D., Jing, Y., Zhao, J., Wang, W., & Song, G. (2017). A fast and efficient algorithm for mining top-k nodes in complex networks. *Scientific reports, 7*, 43330.

Lu, M., Zhang, Z., Qu, Z., & Kang, Y. (2018). LPANNI: Overlapping community detection using label propagation in large-scale complex networks. *IEEE Transactions on Knowledge and Data Engineering, 31*(9), 1736–1749.

Morone, F., Min, B., Bo, L., Mari, R., & Makse, H. A. (2016). Collective influence algorithm to find influencers via optimal percolation in massively large social media. *Scientific reports, 6*, 30062.

Narayanam, R., & Narahari, Y. (2010). A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering, 8*(1), 130–147.

Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E, 67*(2), Article 026126.

Nguyen, D.-L., Nguyen, T.-H., Do, T.-H., & Yoo, M. (2017). Probability-based multi-hop diffusion method for influence maximization in social networks. *Wireless Personal Communications, 93*(4), 903–916.

Qiu, L., Jia, W., Yu, J., Fan, X., & Gao, W. (2019). PHG: A three-phase algorithm for influence maximization based on community structure. *IEEE Access, 7*, 62511–62522.

Roghani, H., & Bouyer, A. (2022). A fast local balanced label diffusion algorithm for community detection in social networks. *IEEE Transactions on Knowledge and Data Engineering, 1–13*. https://doi.org/10.1109/TKDE.2022.3162161

Roghani, H., Bouyer, A., & Nourani, E. (2021). PLDLS: A novel parallel label diffusion and label Selection-based community detection algorithm based on Spark in social networks. *Expert Systems with Applications, 183*, Article 115377.

Rui, X., Meng, F., Wang, Z., & Yuan, G. (2019). A reversed node ranking approach for influence maximization in social networks. *Applied Intelligence, 49*(7), 2684–2698.

Samadi, N., & Bouyer, A. (2019). Identifying influential spreaders based on edge ratio and neighborhood diversity measures in complex networks. *Computing, 101*(8), 1147–1175.

Saxena, R., Kaur, S., & Bhatnagar, V. (2018). Social centrality using network hierarchy and community structure. *Data Mining and Knowledge Discovery, 32*(5), 1421–1443.

Shang, J., Zhou, S., Li, X., Liu, L., & Wu, H. (2017). CoFIM: A community-based framework for influence maximization on large-scale networks. *Knowledge-Based Systems, 117*, 88–100.

Singh, S. S., Kumar, A., Singh, K., & Biswas, B. (2019). C2IM: Community based context-aware influence maximization in social networks. *Physica A: Statistical Mechanics and its Applications, 514*, 796–818.

Taheri, S., & Bouyer, A. (2020). Community detection in social networks using affinity propagation with adaptive similarity matrix. *Big Data, 8*(3), 189–202.

Wang, B., Zhang, J., Dai, J., & Sheng, J. (2022). Influential nodes identification using network local structural properties. *Scientific Reports, 12*(1), 1–13.

Wang, C., Chen, W., & Wang, Y. (2012). Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery, 25*(3), 545–576.

Wang, G., Jiang, J., Li, W., & Wang, C. (2019). *Influence Maximization Based on Node Attraction Model.* Paper presented at the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing. *Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*.

Wei, X., Zhao, J., Liu, S., & Wang, Y. (2022). Identifying influential spreaders in complex networks for disease spread and control. *Scientific reports, 12*(1), 1–11.

Wu, H., Shang, J., Zhou, S., Feng, Y., Qiang, B., & Xie, W. (2018). LAIM: A linear time iterative approach for efficient influence maximization in large-scale networks. *IEEE Access, 6*, 44221–44234.

Xie, G., Chen, Y., Zhang, H., & Liu, Y. (2019). MBIC: A novel influence propagation model for membership-based influence maximization in social networks. *IEEE Access, 7*, 75696–75707.

Zarezadeh, M., Nourani, E., & Bouyer, A. (2021). DPNLP: Distance based peripheral nodes label propagation algorithm for community detection in social networks. *World Wide Web*, 1–26.

Zhang, C., Li, W., Wei, D., Liu, Y., & Li, Z. (2022). Network dynamic GCN influence maximization algorithm with leader fake labeling mechanism. *IEEE Transactions on Computational Social Systems, 1–9*. https://doi.org/10.1109/TCSS.2022.3193583

Zhang, J.-X., Chen, D.-B., Dong, Q., & Zhao, Z.-D. (2016). Identifying a set of influential spreaders in complex networks. *Scientific Reports, 6*, 27823.

Zhang, X., Xu, L., & Xu, Z. (2022). Influence maximization based on network motifs in mobile social networks. *IEEE Transactions on Network Science and Engineering, 9*(4), 2353–2363. https://doi.org/10.1109/TNSE.2022.3163203