Information Sciences 640 (2023) 119105

Contents lists available at ScienceDirect

Information Sciences

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

A fast module identification and filtering approach for influence maximization problem in social networks

Hamid Ahmadi Beni [a], Asgarali Bouyer [a,b,*], Sevda Azimi [a], Alireza Rouhi [a], Bahman Arasteh [b]

[a] Department of Software Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran
[b] Department of Software Engineering, İstinye University, İstanbul, Turkey

ARTICLE INFO

Keywords:
Social networks
Influence maximization
Module detection
Filtering
Independent cascade model

ABSTRACT

In this paper, we explore influence maximization, one of the most widely studied problems in social network analysis. However, developing an effective algorithm for influence maximization is still a challenging task given its NP-hard nature. To tackle this issue, we propose the CSP (Combined modules for Seed Processing) algorithm, which aim to identify influential nodes. In CSP, graph modules are initially identified by a combination of criteria such as the clustering coefficient, degree, and common neighbors of nodes. Nodes with the same label are then clustered together into modules using label diffusion. Subsequently, only the most influential modules are selected using a filtering method based on their diffusion capacity. The algorithm then merges neighboring modules into distinct modules and extracts a candidate set of influential nodes using a new metric to quickly select seed sets. The number of selected nodes for the candidate set is restricted by a defined *limit* measure. Finally, seed nodes are chosen from the candidate set using a novel node scoring measure. We evaluated the proposed algorithm on both real-world and synthetic networks, and our experimental results indicate that the CSP algorithm outperforms other competitive algorithms in terms of solution quality and speedup on tested networks.

1. Introduction

A social network is composed of social actors, such as individuals and organizations, who interact with each other [1]. Social networks have become a platform for Neural Language Generation due to the significant increase in smartphones [2]. However, negative public opinion and the emergence of e-commerce [3] in social networks can lead to significant losses in credit or economic aspects [4]. By understanding cyber activities and analyzing user preferences in social networks [5], useful implications for policy makers can be discovered [6]. Today, most people are members of large-scale online social networks, such as Facebook, Twitter, Instagram, Weibo, WeChat, and LinkedIn, due to the rapid development and application of social networks. These platforms provide a background for detecting community structures [7] and the propagation of information [8,9], making them essential for companies looking to commercialize their products through social networks as a convenient environment for the spread of information between connected people [10]. In social networks, a profitable channel is created for various purposes, such as viral marketing, information campaigns, recommendation systems, and more [11,12]. Viral marketing plays a significant role in the spread of information, ideas,

* Corresponding author at: Department of Software Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran.
E-mail addresses: h.ahmadi@azaruniv.ac.ir (H.A. Beni), a.bouyer@azaruniv.ac.ir (A. Bouyer), s.azimi@azaruniv.ac.ir (S. Azimi), rouhi@azaruniv.ac.ir (A. Rouhi), bahman.arasteh@istinye.edu.tr (B. Arasteh).

https://doi.org/10.1016/j.ins.2023.119105
Received 6 September 2022; Received in revised form 25 April 2023; Accepted 2 May 2023
Available online 8 May 2023

and influences. In viral marketing, a person's decision to buy a product often affects their friends, business partners, and so on [13]. This leads to the emergence of a problem called influence maximization, where the objective is to select a seed set of k nodes in a social network to maximize the influence throughout the network. In other words, the k influential nodes cover the whole network for high information propagation at a low cost. Many algorithms have been proposed in the last decade to improve efficiency, scalability, speedup, and other factors in IMP. For example, some algorithms such as Greedy, CELF (Cost-Effective Lazy Forward selection), CELF++, TSIM (two-stage selection for influence maximization), ASIM (A Scalable algorithm for Influence Maximization), MCDM (Multi-Criteria Decision Making), CoFIM (a Community-based Framework for Influence Maximization) proposed for solving IMP. However, these algorithms suffer from various problems, such as being time-consuming, lacking efficiency and scalability. For example, greedy-based algorithms such as Greedy and CELF++ algorithms cannot provide optimal running time and is not scalable in the large-scale networks. ASIM and MCDM are algorithms that do not have required results in terms of approximation guarantee. CoFIM also suffers from the efficiency problem. For this reason, there is a great significance on proposing new algorithms to tackle the influence maximization problem.

Another important issue in IMP is the used models for information propagation. Three well-known models are used to describe the spread of information between nodes: the independent cascade (IC) model, the linear threshold (LT) model, and the weighted cascade (WC) model. The IC model initiates the diffusion process with a set of active nodes $A0$. When a node v becomes active in step t, it sends an activation signal to each of its currently inactive neighbors, denoted by node w, with a chance of activation only once. If v is successful, then node w becomes active in step $t+1$. In the LT model, each node v selects a random threshold θ_v from the interval [0,1]. Therefore, v is activated when the fraction of its activated neighbors reaches its threshold. Finally, in the WC model, if v is activated in round i, then with probability $1/d_u$, u is activated by v in round $i+1$, where node u has d_u neighbors. Each neighbor activates node u independently, and if node u is currently inactive, then the probability that u is activated in round $i+1$ is $1-\left(1-\frac{1}{d_u}\right)^l$, where l is the number of currently active neighbors of u in round i. The difference between WC and IC models is that the probability of activating u by v is not necessarily the same as the probability of activating v by u. Most recently proposed algorithms for IMP use the IC model to evaluate the algorithm's quality and propagation rate. In this paper, we propose an improved-independent-cascade (IIC) model to evaluate our proposed method and other algorithms on several networks. The IIC model considers the number of activated nodes and the layer of each activated node. Our explorations show that the layer of the activated node is an effective factor in the diffusion process.

The main contribution of this paper is to propose a new algorithm, called the CSP algorithm, for identifying the influential nodes in a network. This algorithm uses the structural features of nodes to generate modules in the network. In the initial phase, each node is labeled based on the graph's topological features. Furthermore, the recognition of modules is done by the label diffusion method [14]. Subsequently, the modules are filtered to reduce the graph search space. Small modules have less chance of activating their neighbors and other modules and are therefore not considered for the candidate node selection step. Moreover, only nodes located nodes in strategic positions inside a module are considered for candidate competition. The number of candidate nodes in each module is determined by the *limit* factor in Eq. (5). Finally, a seed set is chosen from candidate nodes with high importance and better position based on Eq. (6). The performance of the CSP algorithm is compared against several basic and recently proposed algorithms. The CSP algorithm offers several major contributions. First, it presents a new module-based method and a new cascade model to select the best seed nodes, and it performs fast and optimal diffusion on social networks. Second, it reduces the search space by removing modules and fewer important nodes using topological features and the position nodes in networks. Finally, it reduces the overhead of influence diffusion calculations. The performance of the CSP algorithm is compared against several recently proposed and basic algorithms, and the experimental results demonstrate that the proposed method provides superior results than several contemporary methods for influence maximization by optimally identifying the influential nodes in the network.

The rest of the paper is arranged as follows: Section 2 reviews and discusses some related works. The preliminary independent cascade model and the new improved independent cascade model are presented in Section 3. The proposed CSP algorithm is introduced in Section 4. The simulation results and analysis are provided in Section 5. Finally, Section 6 presents the conclusion and future direction.

2. Related works

The problem of social influence maximization was initially proposed as an optimization problem. Two diffusion models, IC (Independent cascade) and LT (Linear threshold), were used to explore its computational cases. These models were used to simulate different diseases like COVID-19, the spreading of advertisements, and so on [15]. The social influence function,$\sigma()$, finds the expected size of the activated set after completion of the diffusion process. It is assumed that the social influence function, $\sigma()$, is submodular and monotone. A set function $f: 2^V \rightarrow R$ is monotone if $f(S) \leq f(T)$, for all $S \subseteq T \subseteq V$. A set function $f: 2^V \rightarrow R$ is submodular if $f(S) + f(T) \geq f(S \cap T) + f(S \cup T)$, for all $S, T \subseteq V$. The proposed solutions to the influence maximization problem can be classified into four categories: greedy-based, heuristic-based, metaheuristic-based, and community-based algorithms. Recently, authors in [16] have comprehensively reviewed and categorized the algorithms for the influence maximization problem.

In the first category, a greedy algorithm is used to find the influential nodes [17]. This algorithm provides $1-1/e$ approximation guarantee on the influence spread, where e is the base of the natural logarithm. The greedy algorithm suffers from two crucial deficiencies: The exact computation of the influence spread, i.e., σ (.), is NP-Complete, and the required number of times to evaluate the influence function, σ (.) is massive. For these reasons, other algorithms have been proposed to improve the greedy algorithm. The main goal of improved algorithms is to reduce the number of simulations. For example, Leskovec et al. proposed the Cost-Effective Lazy

Forward (CELF) algorithm by utilizing submodularity to improve the scalability [18]. CELF significantly decreases the number of times required to evaluate the influence function, σ (.). However, its computational time is still high. The improved CELF, called CELF++, was presented to solve this problem. The key concept of CELF++ is that if node u is included in the seed set in the current iteration, the marginal gain u in σ (.) is not required to be recomputed in the next iteration. The running time of CELF++ is 35–55% faster than CELF. However, computational time is not suitable for large-scale social networks. Based on game theory, Narayam et al. developed the SPIN (ShaPley value based Influential Nodes) strategy [19]. The influence spread is defined using the Shapley value. The nodes are arranged in descending order according to their Shapley values, and the k nodes with the highest Shapley value are chosen as seed nodes. A recent two-stage selection algorithm for influence maximization (TSIM) combines CELF and Discount-degree descending techniques for seed selection [20]. Additionally, a new objective function is defined to increase the accuracy of the selected nodes. This method uses CELF to address degree-discount weakness and uses it to improve CELF performance. However, greedy-based algorithms have limited scalability, which poses a significant challenge. To address this, heuristic algorithms have been proposed. Galhotra et al. proposed a scalable heuristic algorithm for influence maximization under the IC model (called ASIM) [21]. This algorithm assigns a score value for each node (the weighted sum of the number of simple paths starting from the node with at most length d). Compared with CELF++, this algorithm takes less running time and memory as well. These algorithms have better running time and scalability but do not provide any approximation guarantee on the influence spread compared to greedy-based algorithms. The article explores a problem known as Influence Maximization with Information Variation (IMIV) and. Furthermore, Zhang et al. proposed a practical scenario where concepts can stray from their original form to become invalid during message passing [22]. To achieve this, they represent information as a vector and determine the distinction between two arbitrary vectors as a distance calculated by a matching function. As IMIV is NP-hard, the authors choose users in a greedy manner that can approximately maximize the estimation of effective propagation. The DeepIM (Deep learning in Influence Maximization) mechanism was created to address the influence maximization problem [23]. DeepIM determines the most important nodes inside and between networks using local and global structural aspects after using deep learning algorithms to extract the best structural parameters from network nodes. Dai et al. investigated the problem of influence maximization in the diffusion of negative and positive opinions, and their main idea was to maximize the diffusion of positive opinions and minimize the diffusion of negative opinions. For this reason, they have presented a new model based on the independent cascade model, which is the most important factor in the spread of opinions in this model, the group polarization effect [24].The SRFM (Shall-based Ranking and Filtering Method) algorithm is a fast shell-based algorithm that uses a pruning strategy to remove unimportant nodes [25]. The basic idea of the SRFM algorithm is that nodes in highly important shells have different influence power, and also some nodes in less important shells may have high influence power. In this algorithm, nodes located in different shells are chosen as a seed set by considering the role of the bridge nodes inside and outside different shell layers. The authors also proposed an LMP algorithm to select the most influential nodes by leveraging the local traveling and node labeling method [26]. The main feature of this algorithm is its fast running time with acceptable competencies. To address the issues with the greedy algorithm's inefficiency, Li et al. suggested a social network influence propagation model based on the Gaussian propagation model in combination with an influence maximization algorithm based on the same model. The efficiency may be increased using the influence maximization algorithm, which is based on the Gaussian propagation model [27]. Rao et al., based on the fact that in today's world, a new concept called group buying has been proposed in business platforms. The purpose of selecting seed nodes in this paper is to maximize the number of group members by social influence, which selects seed nodes using an adaptive greedy algorithm [28].

On the other hand, Metaheuristic-based algorithms have been proposed to solve the social influence maximization problem using various metaheuristic methods. Bucur et al. were the first to use a genetic algorithm to find an approximate solution for influence spread within a practical running time [29]. Their method is comparable with the basic greedy algorithm [30]. Additionally, Tsai et al. developed a combination of genetic and greedy algorithms under the IC model for the influence maximization problem, called a genetic new greedy algorithm [31]. Experimental results showed that the genetic new greedy algorithm was 10% better than the genetic algorithm in terms of influence spread. Furthermore, a multi-criteria decision-making integrated adaptive simulated annealing approach was proposed for the influence maximization problem [32]. For competitive models, Liang et al. colleagues provide a uniform function based on the pruning algorithm to estimate the spread of influence [33]. The MCDM algorithm excludes less influential nodes to reduce costs and ranks nodes based on the centrality measure. In the next step, a self-adaptive degree mechanism has been introduced to increase performance in the creation and evaluation process. To speed up the convergence process, a greedy hill-climbing strategy has been applied. The obtained results show that the running time of this algorithm is better than the previous methods. The main advantage of these algorithms is their acceptable running time, but the disadvantage is that they do not provide an approximation guarantee. Jabbari et al. proposed a new approach to influence maximization problem by using dynamic generalized genetic algorithm under IC models to identify a dynamic seed set [34]. Kumar et al., to improve the spread of influence by seed nodes, first use struc2vec to extract features for the basis of the graph structure for each node, then selects seed nodes using Graph Neural Network [35]. DDSE (Degree-Descending Search Evolution) [36] is a new evolutionary-based algorithm that uses the degree-descending search strategy and an extended evolutionary algorithm to improve performance and eliminate overhead-time, which is often encountered in greedy algorithms.

The fourth and last category is *community-based* algorithms which are more efficient than the previous algorithms in terms of scalability and diffusion capabilities. The community structure is reflected in a real-life social network [37], where nodes are dense and sparsely connected with the outside nodes [38]. Wang et al. proposed the community-based greedy algorithm, which finds communities based on information propagation and selects influential nodes using communities [39]. The TI-SC (Top-k influential nodes selection based on community detection) algorithm is another community-based heuristic algorithm that selects efficient seeds based on community structures and local topologic information [40]. This algorithm calculates a score for each node based on the score of 1-hop and 2-hop neighbors, and the highest-scored node is defined as the first seed. When a node is selected as a seed, the scores of other

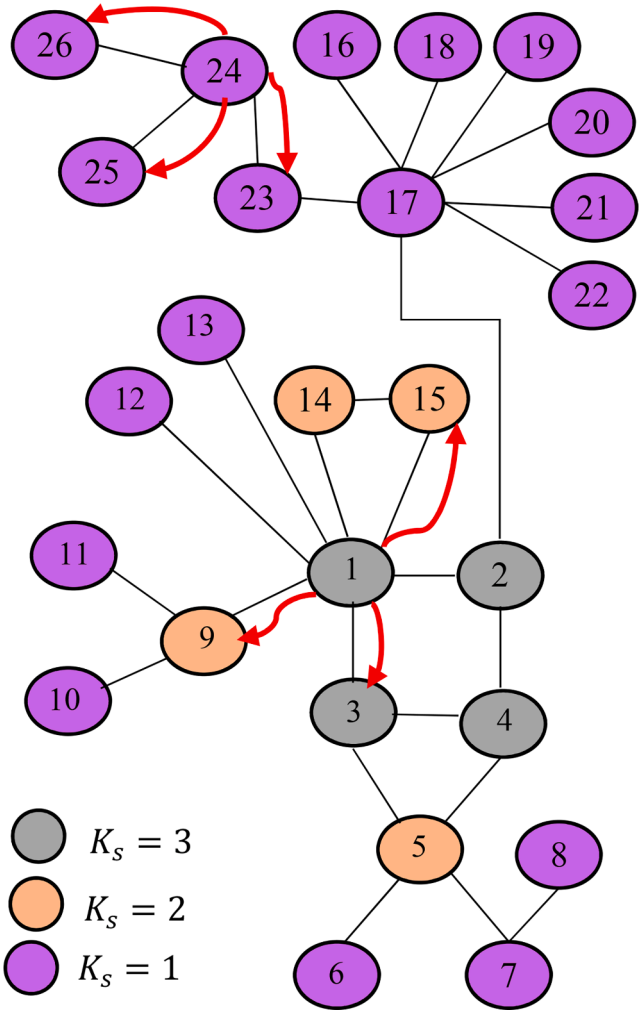


Fig. 1. An example of calculating a new influence expansion criterion.

nodes are updated to minimize possible overlap. Shang et al. proposed CoFIM, a two-phase propagation model that expands the seed set to the neighbor nodes of S in the first phase and computes the influence spread within communities in the second phase. This algorithm achieves better influence spread compared to previous methods [41]. In terms of influence spread, this algorithm is similar to the CoFIM method but more efficient. The FIP method (Fast Influential Nodes Processing for IM problem), introduced in 2023, chooses seed nodes based on the precedence of overlapping and non-overlapping nodes [42]. It calculates the likelihood of spreading inside and outside the community based on the nodes' tactical placement. But it cannot select seed nodes very fast in large-scale networks. Under an independent cascade model, the IMBC algorithm was suggested as a solution to the influence maximization problem. The steps in this algorithm are as follows: Communities are first identified in the initial phase, then converted into a supergraph so that communities are considered supergraph nodes and the links between communities are considered supergraph edges. Then, communities are selected for diffusion based on the characteristics of the set of minimum dominating nodes in the supergraph [43]. The CBIM algorithm has a stage for selecting seed nodes, and in the first stage, communities are identified in multilayer networks by using the similarity between neighbors. Then, based on the degree and distance of the node, a weight is assigned to each node. Then, in each community, seed nodes are selected based on the community structure [44].

3. The improved independent cascade model

The independent cascade (IC) model is a mathematical model used to describe the random diffusion of information [45]. In this model, node $v_1 \in V$ initially is an active social entity and its neighbor $v_2 \in V$ is an inactive node. Accordingly, node v_1 may activate its inactive neighbor v_2 based on the probability propagation $p(v_1,v_2)$ on the edge $e = (v_1,v_2) \in E$. Each node in the IC model has two modes: (i) *active*, which means a node has already been affected by the information being disseminated, and (ii) *inactive*, which means a node is unaware of the information or is not affected. The activation process is performed in discrete steps. At the beginning of

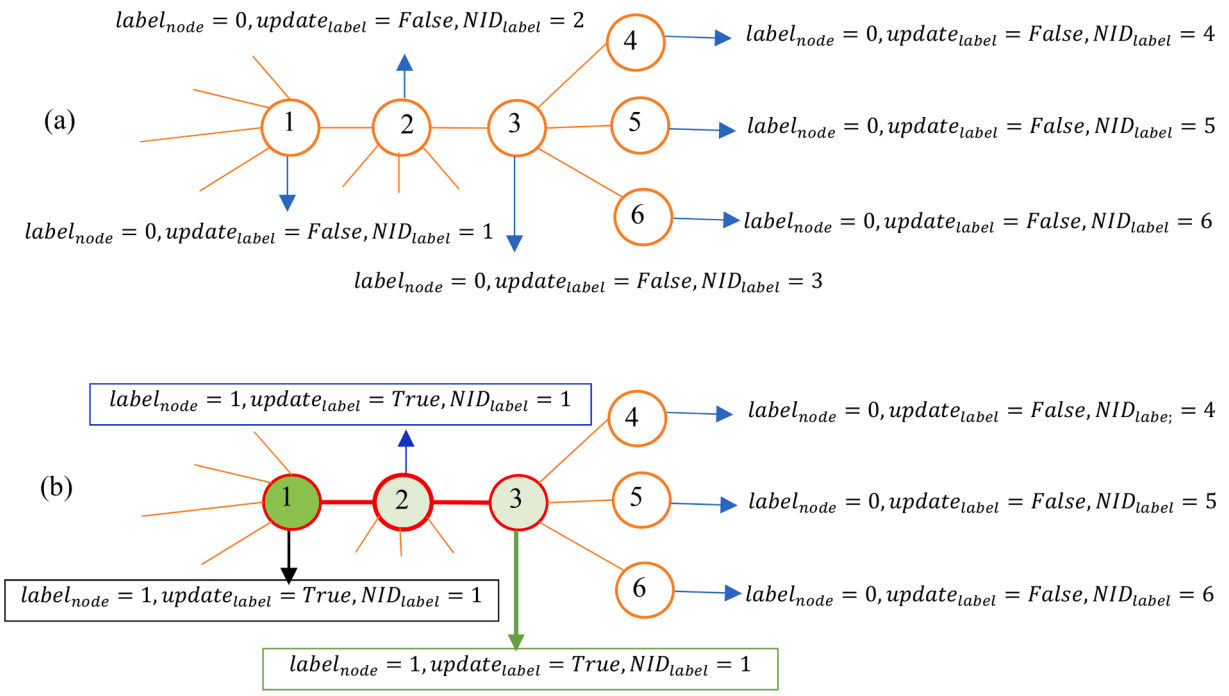


Fig. 2. An example to understand the 3-tuples and their values.

processing IC model, a small number of nodes are given the information known as seed nodes (initially active nodes). Upon receiving the information, these nodes are activated. At each stage, an active node tries to influence its inactive neighbors. Regardless of its success, that node will never have another chance to activate the same inactive neighbor. This process ends when there are no more nodes to try the activation test. In the IC model, the diffusion expansion is based on the number of activated nodes. But the number of activated nodes is not a suitable criterion for calculating real-world diffusion. For example, suppose there are two nodes: the first node is a student role, and the other node is a professor at a university. Both of them activate 50 nodes to diffuse new information. However, is the extension of the influence equal for these two nodes? In the real world, the professor definitely activates more powerful 50 nodes in terms of diffusion process than the student. In social networks, this issue can be tested by examining the layers of the underlying network. The graph layers play a critical role in the influence spreading on social networks. Social network layering is done by the k-shell algorithm. In the k-shell algorithm, the nodes with degree 1 are first removed from graph G and placed in shell 1. After removing the current degree-1 nodes and their connected edges, other degree 1 nodes will possibly be produced in the graph. These nodes are also removed and placed in shell 1. The k-shell algorithm continues until no degree-1 nodes remain in the graph. Degree-2 nodes are then removed and placed in shell 2. This process may create new degree-2 nodes that are also placed in shell 2 until there are no more such nodes in the graph. Ultimately, the algorithm terminates when no more nodes remain in the graph.

On the other side, Influential nodes are usually found in the innermost layers, such as the core layer, or in layers close to the core. Conversely, nodes with low influence tend to reside in the outer layers or periphery layers. To account for the shell number's effect on activated nodes, a new criterion for calculating the diffusion process is proposed. In addition to the number of activated nodes, the shell number for each activated node is also considered. For example, if node a has activated three nodes in the periphery shells and also node b has activated three nodes in the core shell, the diffusion rate of nodes a and b must be distinguished. Nodes activated in the core shell have a high diffusion rate, resulting in optimal information spreading on social networks. Thus, the new criterion for the influence expansion $IS_{N_{v_i}}$ is defined according to Eq. (1):

$$IS_{N_{v_i}} = \sum_{i \in l} K_{s_i} \times n_{a_i} \tag{1}$$

Where K_{s_i} and n_{a_i} are the shell number and the number of activated nodes by the node v_i in shell i, respectively, and also l is the number of the network's shells.

Fig. 1 depicts the graph shells with different colors, including gray nodes in shell 3, orange nodes in shell 2, and purple nodes in shell 1. As shown in the figure, both nodes 24 and 1 have activated 3 nodes, but the positions of the activated nodes are completely different. In other presented algorithms, these two nodes have the same influence because they have activated the same number of nodes. While according to the criterion $IS_{N_{v_i}}$, node 1 has more influence expansion than node 24 because $IS_{N_{24}} = 3 \times 1 = 3$ and $IS_{N_1} = (1 \times 3) + (2 \times 2) = 7$.

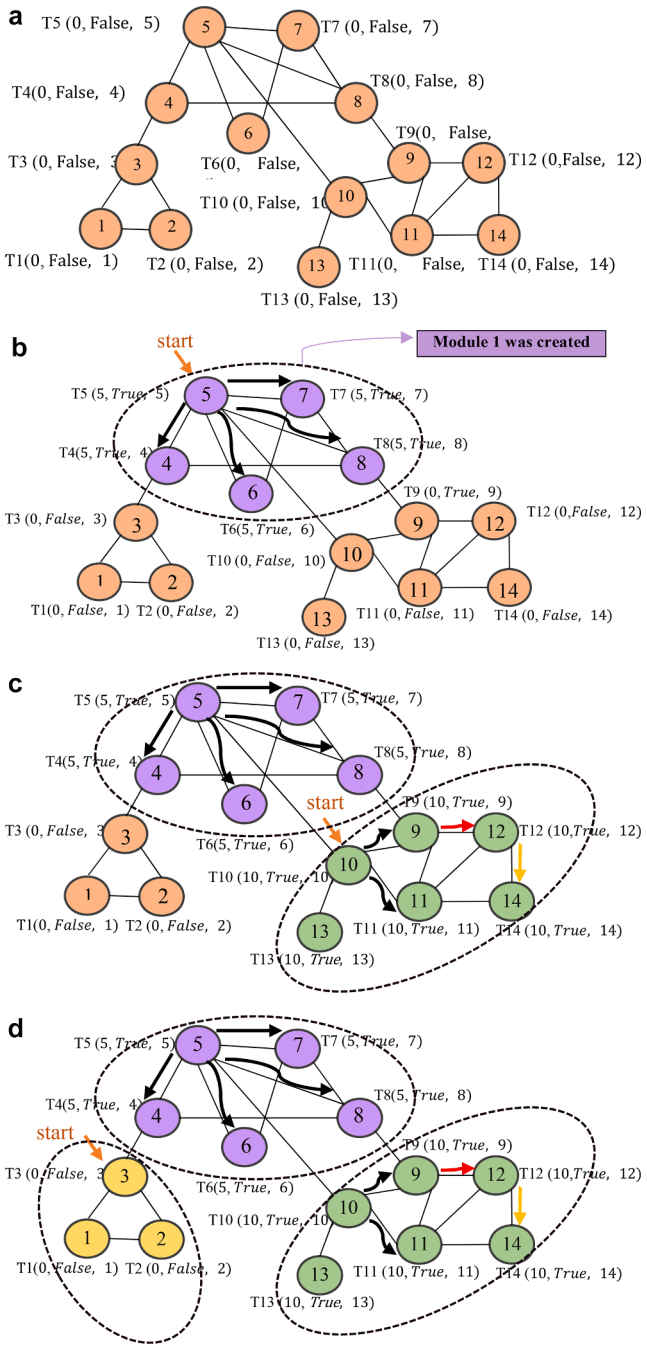


Fig. 3. The module identification in an example network.

4. Proposed algorithm

The proposed algorithm for detecting highly influential nodes follows four steps: module identification, module filtering, candidate node selection, and selection of final seed nodes. These steps are described in detail below.

4.1. Module identification

First, modules are identified by analyzing the network structure. Each module consists of a group of nodes with tightly-coupled relations inside the group. A module is not necessarily a community structure, but it may be a part of a community. At first, the 3-

Table 1
Graph information based on degree in descending order & Clustering Coefficient in ascending order.

ID	Degree	Clustering Coefficient
5	5	0.3
10	4	0.16
8	4	0.33
9	4	0.33
11	4	0.5
12	4	0.66
3	3	0.33
4	3	0.33
7	3	0.66
1	2	1
2	2	1
6	2	1
14	2	1
13	1	0

tuple $L(label_{node}, update_{label}, NID_{label})$ is defined for each node in the network. $label_{node}$ indicates the label of each node that initially is set to 0 for all nodes before detecting the modules. In other words, each node with $label_{node} \rightarrow 0$ has not previously been checked to add a module. The $label_{node}$ is updated based on the two variables $update_{label}$ and NID_{label}. Eventually, nodes with the same value of $label_{node}$ are placed in a module. The initial value for NID_{label} is the node index. In the label diffusion process, if a traversal is performed from node x to node y, then NID_{label} of node y holds the NID_{label} label of node x. Also, the initial setting value for $update_{label}$ is equal to *False*, which is used for updating labels. If the traversal is performed from node x to node y and if the clustering coefficient of node y is equal to or greater than the clustering coefficient of node x, then the value of the variable $update_{label}$ changes from *False* to *True*. Thus, the node label is updated based on NID_{label}. Fig. 2 helps to better understand this process and the variables in the tuple.

In Fig. 2.a, the labels are assigned before label propagation with a default value of 0, i.e., $label_{node} \rightarrow 0$, and the value of $update_{label}$ is set to *False* for all nodes. The value of NID_{label} for each node is based on the node index. For example, node 6 has NID_{label} of 6. Fig. 2.b illustrates how the labels are updated after propagation. If node 1 is selected as the starting node for propagation, its $label_{node}$ value changes from 0 to 1, and its $update_{label}$ value changes from *False* to *True*. Now, Node 2 is selected as the target node for propagation, and since its clustering coefficient is assumed to be higher than that of node 1, its $NID_{label} = 1$ and also the value of $update_{label}$ is also set to *True*. Next, because the value of the variable $update_{label}$ for node 2 was set to *True*, then its label is updated based on NID_{label} value, which is 1. In other words, $label_{node} = NID_{label}$, i.e., $label_{node\ 2} = 1$. Node *2* then selects node *3* for label diffusion and since its clustering coefficient is assumed to be higher than that of node *2,* so for node *3*, NID_{label} *of node 3* $= NID_{label}$ *of node 2*. Thus, NID_{label} *of node 3* $= 1$ and $label_{node\ of\ node\ 3} = 1$, and also, the variable of $update_{label}$ is changed from *False* to *True*. Next, because for node 3 the value of the variable $update_{label}$ has been changed from *False* to *True*, then the node label is updated based on NID_{label}, in other words, $label_{node\ 3} = NID_{label}$, i.e., $label_{node} = 1$. Now, node 3 can select nodes 4, 5, and 6 for label diffusion. The same process continues for nodes 4, 5, and 6, but since their clustering coefficients are assumed to be lower than node 3, their variables $label_{node}$, $update_{label}$, and NID_{label} values remain the same.

To identify the modules, a list of nodes is constructed in descending order based on their degree, and this list is called N_{sort}. If two nodes have the same degree, a node with a smaller clustering coefficient is selected for traversal. The node clustering coefficient is also used in the label diffusion process. The first target node is selected from the list N_{sort}, and its initial label is updated, and the value of $label_{node}$ changes to its NID_{label} value. The label diffusion process is performed up to three levels of neighboring nodes. If the target node x has been selected in the first level for traveling, then any neighbor node of node x that has a clustering coefficient greater or equal than node x (such as neighbor node y), the value of $update_{label}$ for node y changes from *False* to *True*. Also, the value of NID_{label} for node y changes to the value of $label_{node}$ node x. Then, the value of $label_{node}$ for node y changes to the NID_{label} value. If any neighbor node of node x has a lower clustering coefficient than node x, the variables $update_{label}$ and NID_{label} for these nodes are not changed. In the second level of traveling from target node x, the label propagation conditions from the updated node y to its neighbors (such as neighbor node z) are slightly different. If node z has a clustering coefficient greater than node y, and the degree of node y is greater than node z, and they have at least one common neighbor, then the value of NID_{label} for node z changes to $label_{node}$ of node y. Next, the value of $label_{node}$ for node z changes to the value of NID_{label}. If any neighbor node of node y has not these conditions, the variables $update_{label}$ and NID_{label} are not changed. The process described above is also applied to nodes in the third level of the target node x, based on defined conditions for the second level of neighbors. Once these conditions are met, diffusion is terminated, and the desired module is selected. To construct the next modules, the above steps are repeated for each unselected node with $update_{label} = False$. The module construction ends when no node has a $label_{node}$ value of 0. Moreover, degree-1 nodes do not undergo any traveling process. Instead, they belong to the module of their only connected neighbor and are placed in the same module as that neighbor.

Example 1. To illustrate the module construction process, consider the example in Fig. 3. In this example, we start with an initial graph and a table that lists all the nodes and their information (Table 1). The module construction process begins by selecting the first node in the table (node 5) as the starter or target node. Then, the initial label of this node is changed to $label_{node5} = 5$, and also the variable $update_{label}$ is also changed to *True* due to the change of label. The label can be propagated up to 3 levels in the graph. In the first

Table 2
Some statistical properties of the applied datasets: node number |V|, Edge number |E|, Average Degree, and Maximum degree.

Networks	Nodes	edges	Average Degree	Maximum degree
Route views	6474	13,895	4.292 55	1459
As22july	22,693	48,436	2.1093	2390
Cond-mat	23,133	93,497	8	279
Douban	154,908	327,162	4	287
Hamster	1858	12,534	13.491 9	272
RoadNet_PA	1,088,092	1,541,898	2	9
Deezer Europe	28,281	92,752	6.55	175
Gr_Qc	5242	14,496	5	81

level of label propagation, all neighbor nodes of node 5 that have a clustering coefficient greater than or equal to node 5 (nodes 4, 6, 7, and 8) are selected, and the value of $update_{label}$ of these nodes are changed from *False* to *True*. Also, the value of NID_{label} for these nodes is set to the value of $label_{node}$ of node 5. This label changing has been indicated by coloring in Fig. 3(b). Next, nodes 8 and 4 correspondingly continue the propagation process. However, the label diffusion cannot be continued to nodes 3 and 9 because the required conditions are not met. In this way, the first module is identified which is colored in Fig. 3(b). To identify the second module, we select the second node from the table (node 10) since it has the lowest clustering coefficient amongst nodes with degree 4. $label_{node} = NID_{label}$ (i.e., $label_{node} = 10$), and the value of $update_{label}$ for this node is changed to *True*. The label diffusion in the first level is carried out from node 10 to nodes 9 and 11. The $label_{node}$ values of nodes 9 and 11 are changed from 0 to 10 and the $update_{label}$ value of these nodes are set to *True*. Based on the diffusion conditions mentioned in the second level, propagation from node 9 to node 12 is possible. Therefore, $label_{node}$ node 12 is changed from 0 to 10 and the value of $update_{label}$ becomes *True*. To do the diffusion in the third level from node 12 to 14, the previous conditions are checked again. Once the conditions are met, the label diffusion is performed and $label_{node}$ of node 14 is changed from 0 → 10 and the value of $update_{label}$ becomes *True*. The diffusion process is stopped, and the second module is selected. Node 13, which has degree 1 and is connected to node 10 in the second module, also receives the label of the second module, and its values are updated. We explore the table list and skip the nodes that have already been placed in the modules. Finally, node 3 is selected as the target node, and the first level label propagation is carried out for nodes 1 and 2. The third module is then constructed. The process is repeated for each unselected node with $update_{label} = False$ until there are no nodes left with $label_{node} = 0$. It can be seen that no nodes have been left unexplored, and the tuples of all nodes have been updated. Thus, the module creation process is terminated.

4.2. Module filtering and candidate nodes selection

In the influence maximization problem, the main challenge lies in the search space. Most previous algorithms for this problem are not adequately scalable for medium or large-scale networks. One of the primary goals of the CSP algorithm is to find a seed set in a linear time complexity. Therefore, after identifying modules, the module filtering method is used to reduce the graph search space. Modules with a smaller size (in terms of the number of nodes included) have a lower chance of activating the remaining modules. Therefore, it is necessary to effectively filter out smaller and less influential modules. In this section, modules whose size (n_{c_i}) is smaller than the θ threshold will be removed. In other words, the θ criterion serves as a decision point for selecting appropriate modules. The θ threshold is computed using Eq. (2), as follows:

$$\theta = \frac{\sum_i n_{c_i}}{len(c)} \tag{2}$$

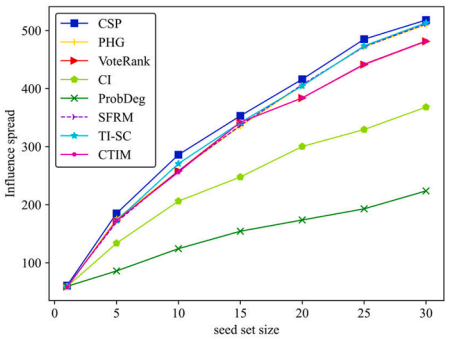
Eq. (2) defines n_{c_i} as the size of module c_i (the number of nodes in the module), and *len(c)* as the total number of modules in the graph resulting from the previous step. Tightly connected modules are then merged. Specifically, given two neighboring modules C_i and C_j, if C_i is smaller, it can be merged with C_j if the following condition holds, according to Eq. (3):

$$\frac{n_{C_i}}{n_{C_j}} \times Ein_{C_i}) - E_{out}(C_i, C_j) \leq 1 \tag{3}$$

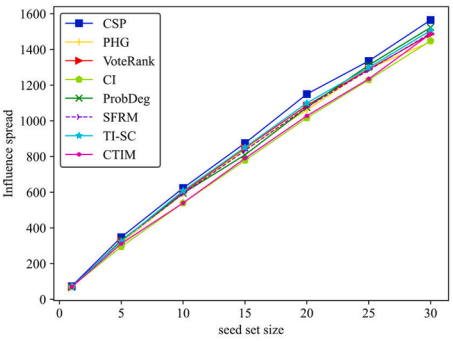
Here, n_{c_i} and n_{c_j} are the numbers of nodes in modules c_i and c_j, respectively. Also, Ein_{C_i} is the number of edges in module c_i, and $E_{out}(C_i, C_j)$ is the number of edges between nodes in the module C_i and C_j. Next, the filtering method is applied within each module to remove less important nodes. To identify the high-importance nodes, filtering is only performed on the remaining modules. The node influence rate p_{v_i} is computed to filter nodes in a module based on their degree, clustering coefficient, and common neighbor, using Eq. (4):

$$p_{v_i} = \left(\frac{\sum_{v_j \in N_e} d_{v_i} * (d_{v_j} - 1)}{\mathrm{d_{max}}}\right) - \left(R_{v_i} * \sum_{v_j \in N_e} (N_{v_j} \cap N_{v_i})\right) \tag{4}$$

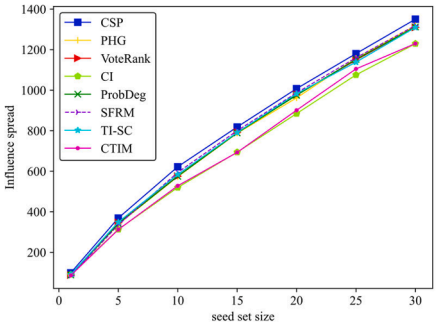
In Eq. (4), N_e represents the neighbors of node v_i, R_{v_i} represents the clustering coefficient of node v_i, and N_{v_j} and N_{v_i} are the neighbors of nodes v_i and v_j, respectively. d_{max} is the largest degree in the module where the node is located. The first part of Eq. (4) aims to identify nodes and their neighbors that have favorable degrees. The second part aims to identify nodes that are not in dense



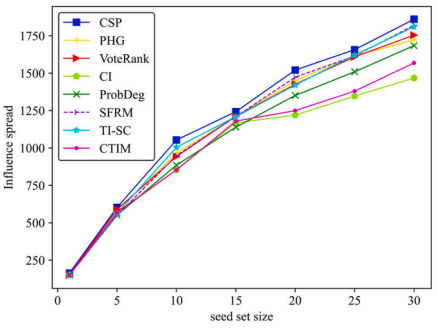
a.Route views



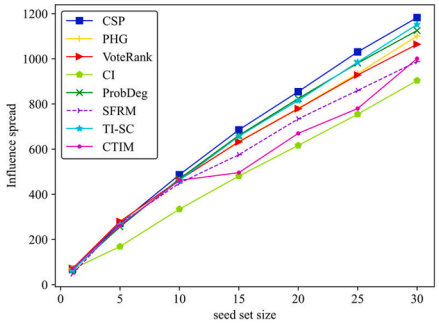
b.Douban



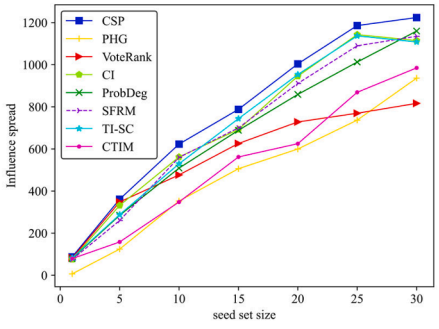
c.Hamester



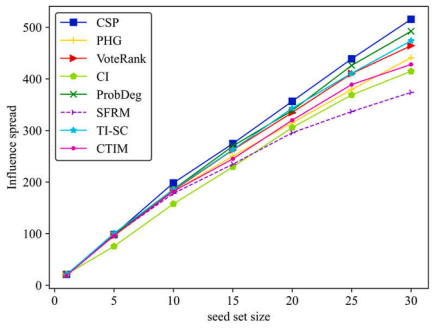
d.as22july



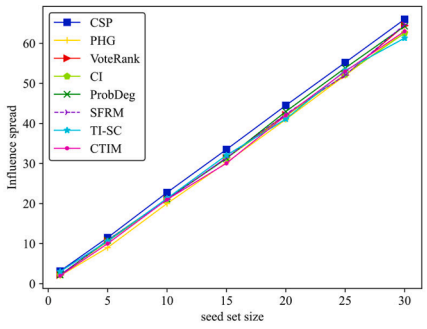
e. Cond-mat



f. Deezer Europe



g. RoadNet_PA



h. Gr_Qc

(caption on next page)

Fig. 4. Influence spread comparison in 8 real-world benchmark datasets under the IIC model.

points (e.g., k-cliques or k-plex) but are well positioned in terms of degree and the importance of their neighbors, and have fewer common neighbors (e.g., bridges). These calculations are performed on all nodes in a module, and the nodes are then sorted in descending order based on the p_{v_i} values. Finally, Eq. (5) that is called *limit* equation, is used to determine the limited number of candidate nodes in each module. This equation computes the number of nodes that have the best chance of diffusion in a module, based on their p_{v_i} values as follows:

$$limit = 2k\frac{X_{Ci}}{X_{MAX}} \tag{5}$$

In Eq. (5), k is the number of required seed nodes in the next step and X_{Ci} is the total number of nodes in a module Ci and X_{MAX} is the number of nodes in the largest module.

4.3. Selection of seed nodes

Candidate nodes are selected using the method introduced in the previous step. In this step, seed nodes are chosen from the candidate nodes and added to the final *seed set*. The IP_{v_i} measure presented in Eq. (6) is used for seed node selection. The IP_{v_i} measure utilizes the degree of node v_i ($\deg_{v_i}$), the PageRank of node v_i (p_{v_i}), and the number of independent paths (I_{v_i})between v_i and v_j where v_j is the largest degree node up to distance 2 from node v_i. Eq. (6) represents the calculation of the IP_{v_i} measure.

$$IP_{v_i} = \left(deg_{v_i} + I_{v_i}\right)^* p_{v_i} \tag{6}$$

During the seed node selection step, it was discovered that high-degree nodes may not be optimal for information propagation. To address this, the local criteria p_{v_i} and I_{v_i} are utilized to select he most suitable seed nodes. These criteria help identify candidate nodes with powerful neighbors for efficient information diffusion. Additionally, a higher number of independent paths from a seed node to the highest degree node up to the second level of neighbors indicates that the chosen seed node has a better chance of activating the higher degree node using different paths in its 1-hop or 2-hop neighborhood. After calculating the IP_{v_i} criterion for all candidate nodes, the k nodes with the highest value of IP_{v_i} are selected as the seed. This selection process ensures that the seed nodes are capable of efficiently propagating information throughout the network, leading to a successful diffusion process.

5. Experiments and evaluation

In this section, experiments are conducted on real-world social networks to validate the effectiveness and efficiency of the CSP algorithm. First, the benchmark networks and compared algorithms are introduced. After that, experiments are conducted to determine the influence spread rate with different seed sizes. Then, the CSP algorithm is compared with other state-of-the-art algorithms to evaluate its performance. Finally, we compare and discuss the experimental results of all algorithms based on the diffusion model and their run time.

5.1. Datasets

Eight real networks are used to test CSP in dealing with the problem of influence maximization. The statistical characteristics of these networks are shown in Table 2. The brief descriptions for these networks are presented as follows.

- *Cond-mat* [46]: A co-authorship network of scientists working on Condense Matter Physics.
- *Hamster* [47]: A friendship network among users of the website hamsterster.com.
- *As22july* [47]: This network is in the collection of Miscellaneous Networks.
- *Route views* [47]: This is the undirected network of autonomous systems of the Internet connected with each other. Nodes are autonomous systems (AS), and edges denote communication.
- *Douban* [48]: It is an online social network providing user review and recommendation services for movies, books, and music. The dataset contains all links among users.
- *Deezer Europe* [6]: A social network of Deezer users which was collected from the public API in March 2020. Nodes are Deezer users from European countries and edges are mutual follower relationships between them.
- *RoadNet_PA* [49]: This is a road network of Pennsylvania. Intersections and endpoints are represented by nodes, and the roads connecting these intersections or endpoints are represented by undirected edges.
- *Gr_Qc* [49]: The collaboration network is from the e-print ArXiv and covers scientific collaborations between author's papers submitted to General Relativity and Quantum Cosmology categories.

5.2. Influence spread comparison

In this section, we present extensive experiments conducted on eight social networks to further demonstrate the performance of the CSP algorithm [50]. For all experiments, the number of seed sets k was varied from 1 to 30, and the Monte-Carlo simulation numbers

Table 3
The *Speedup* ratio of the CSP algorithm over other algorithms.

Data sets	Seed size	**Algorithms**													
	k	CTIM	CSP	TI-SC	CSP	SFRM	CSP	PD	CSP	CI	CSP	VR	CSP	PHG	CSP
As-22july1	10	−18.9	23.3	−4.6	4.8	−9.8	10.9	−15.9	18.9	−19.0	23.5	−10.5	11.7	−8.4	9.2
	20	−17.8	21.6	−6.5	7.0	−3.1	3.2	−11.1	12.5	−19.7	24.6	−6.2	6.6	−4.5	4.8
	30	−15.6	18.6	−2.6	2.7	−2.1	2.2	−9.4	10.4	−21.1	26.7	−5.7	6.0	−7.4	8.0
Cond-mat-1995	10	−5.0	5.3	−5.0	5.3	−7.6	8.2	−3.7	3.8	−31.3	45.6	−4.6	4.9	−4.6	4.9
	20	−21.6	27.5	−4.5	4.7	−14	16.3	−3.6	3.7	−27.8	38.5	−8.8	9.6	−8.6	9.4
	30	−15.3	18.15	−2.6	2.6	−16.4	19.7	−4.8	5.1	−23.5	30.8	−10.0	11.1	−6.9	7.4
Deezer-Europe	10	−8.1	8.9	−6.4	6.8	−10.2	11.4	−6.2	6.6	−20.4	25.7	−7.3	7.9	−8.2	9.0
	20	−10.2	11.1	−3.7	3.8	−17.0	20.6	−5.0	5.2	−14.2	16.6	−6.1	6.4	−11.6	13.1
	30	−16.9	20.4	−8.0	8.8	−27.4	37.9	−4.5	4.7	−19.5	24.3	−9.9	11.0	−12.7	14.5
	10	−13.4	15.4	−2.5	2.5	−5.0	5.3	−4.7	4.9	−13.4	15.5	−3.8	4.0	−5.1	5.4
Douban	20	−10.6	11.8	−4.3	4.5	−6.3	6.8	−6.5	6.9	−11.6	13.1	−5.5	5.8	−7.7	8.4
	30	−4.9	5.1	−3.7	3.8	−4.6	4.9	−2.6	2.7	−7.4	8.0	−5.0	5.3	−7.7	8.4
	10	−44.1	79.0	−15.0	17.6	−10.1	11.2	−18.2	22.3	−9.8	10.8	−23.4	30.7	−43.3	76.6
Gr-QC	20	−37.7	60.6	−5.0	5.3	−9.2	10.2	−14.4	16.8	−5.8	6.1	−27.4	37.8	−40.2	67.4
	30	−19.5	24.2	−9.4	10.4	−7.2	7.8	−5.2	5.5	−8.9	9.7	−33.2	49.8	−23.5	30.7
	10	−15.1	17.7	−6.1	6.5	−4.4	4.7	−7.5	8.1	−16.4	19.6	−7.4	8.0	−7.8	8.5
Hamester	20	−10.5	11.8	−2.4	2.5	−2.0	2.0	−3.2	3.3	−12.2	13.9	−2.9	3.0	−4.5	4.7
	30	−14.0	16.2	−8.4	9.2	−7.7	8.4	−8.2	9.0	−14.0	16.3	−8.4	9.1	−7.8	8.5
	10	−7.6	8.3	−6.5	6.9	−7.4	8.0	−7.2	7.7	−7.7	8.3	−7.4	7.9	−12.0	13.7
RoadNet_PA	20	−5.6	5.9	−7.6	8.2	−4.8	5.1	−3.4	3.5	−5.9	6.2	−5.3	5.5	−7.8	8.5
	30	−4.5	4.7	−7.1	7.6	−2	2.0	−2.6	2.6	−5.2	5.5	−2.3	2.4	−4.5	4.7
	10	−10.4	11.7	−5.3	5.6	−9.6	10.7	−56.5	130.0	−27.9	38.7	−9.7	10.8	−9.7	10.8
Route views	20	−7.6	8.3	−2.7	2.8	−2.2	2.2	−58.2	139.5	−27.8	38.5	−7.7	8.4	−2.5	2.6
	30	−6.9	7.4	−0.9	0.9	−1.3	1.3	−56.8	131.5	−28.9	40.7	−7.0	7.6	−1.5	1.5

Table 4
The average influence spread of the algorithms.

Algorithms	Route views	RoadNet-PA	Hamester	Gr-QC	Douban	Deezer- Europe	Cond-mat-1995	As-22july1
PHG	391.13	41.3	950.97	629.57	1031.82	315.61	782	1379.10
VR	374.35	42.54	951.82	673.84	1057.16	327.59	769.21	1374.52
CSP	406.66	44.41	1019.95	950.33	1113	356.77	841.40	1478.51
CI	291.45	41.80	877.89	874.13	1001.69	292.67	618.31	1180.39
PD	173.89	42.79	954	842.63	1064.22	338.92	805.87	1307.06
SFRM	392.03	42.68	966.66	868.66	1053.52	282.44	723.97	1414.47
TI-SC	396.15	41.21	958.69	863.63	1071.56	334.16	809.89	1412.23
CTIM	374	42	886.33	652.66	1018.66	310	711	1224.33

used to measure the influence of the seed set were set to 1000 rounds. The influence probability p was set to 0.01 for all tests. The results of the influence spread of the eight algorithms under the IIC model are shown in Fig. 4. The results in Fig. 4 indicate that the CSP algorithm outperforms other algorithms in terms of influence spread. It achieves a promising influence spread rate and a stable rising marginal gain with an increase in seed size, indicating its robustness in finding influential nodes.

In Fig. 4(a) (Route views dataset), the CSP algorithm has the best influence spread, followed by TI-SC in the second rank, which is better than other algorithms. PHG and SFRM achieve less influence spread compared to CSP. VoteRank and CTIM are ranked at a medium level among these algorithms. The CI and ProbDeg algorithms have relatively poor performance, but CI performs better than ProbDeg.

In Fig. 4(b) (Douban), (c) (Hamster), and (d) (as22july), the CSP algorithm shows the best performance compared to other algorithms. In Fig. 4(b), the influence spread of SFRM is comparable to the PHG, VoteRank, and ProbDeg. Additionally, in Fig. 4c, it overcomes other methods. CTIM and CI exhibit poor performance in Fig. 4(b), (c), and (d).

In Fig. 4(e) (Cond-mat) and (f) (Deezer Europe), the CSP algorithm still outperforms other algorithms. In Fig. 4(e), TI-SC shows better performance than ProbeDeg, PHG, VoteRank, and SFRM. In Fig. 4(f), ProbDeg performs better than PHG, VoteRank, and CTIM.

In Fig. 4(g) (RoadNet_PA), the influence spread of all eight algorithms increases constantly with seed nodes, and the influence spread of CSP is larger than other compared methods. TI-SC's influence spread is comparable to that of PHG, VoteRank, and CI, but it yields to SFRM and ProbeDeg. CTIM achieves a larger influence spread than TI-SC when the number of seed nodes is 20 and 30.

Table 3 presents the *Speedup* comparison of the CSP algorithm with other algorithms in terms of influence spread. The VoteRank algorithm and ProbeDegree algorithm are abbreviated as VR and PD, respectively. The *Speedup* x to y is computed using Eq. (7), where x represents the influence spread of the CSP algorithm, and y represents the influence spread of other algorithms for different seed node values. To calculate the *Speedup* ratio x to y, the positions of x and y in Eq. (7) are swapped. The results indicate that the CSP algorithm achieves the highest *Speedup* ratio compared to other algorithms in all examined networks.

$$Speedup_{x\to y} = \left(\frac{(x-y)}{y}\right)*100 \tag{7}$$

The results presented in Table 3 demonstrate that the CSP algorithm outperforms other algorithms in terms of speedup measures in all networks. In Table 4, the average influence spread of the algorithms was evaluated for k = 10, 20, and 30 seed nodes using different datasets. The results of these calculations indicate that the proposed CSP algorithm exhibits the best performance among all compared algorithms.

5.3. Running time comparison

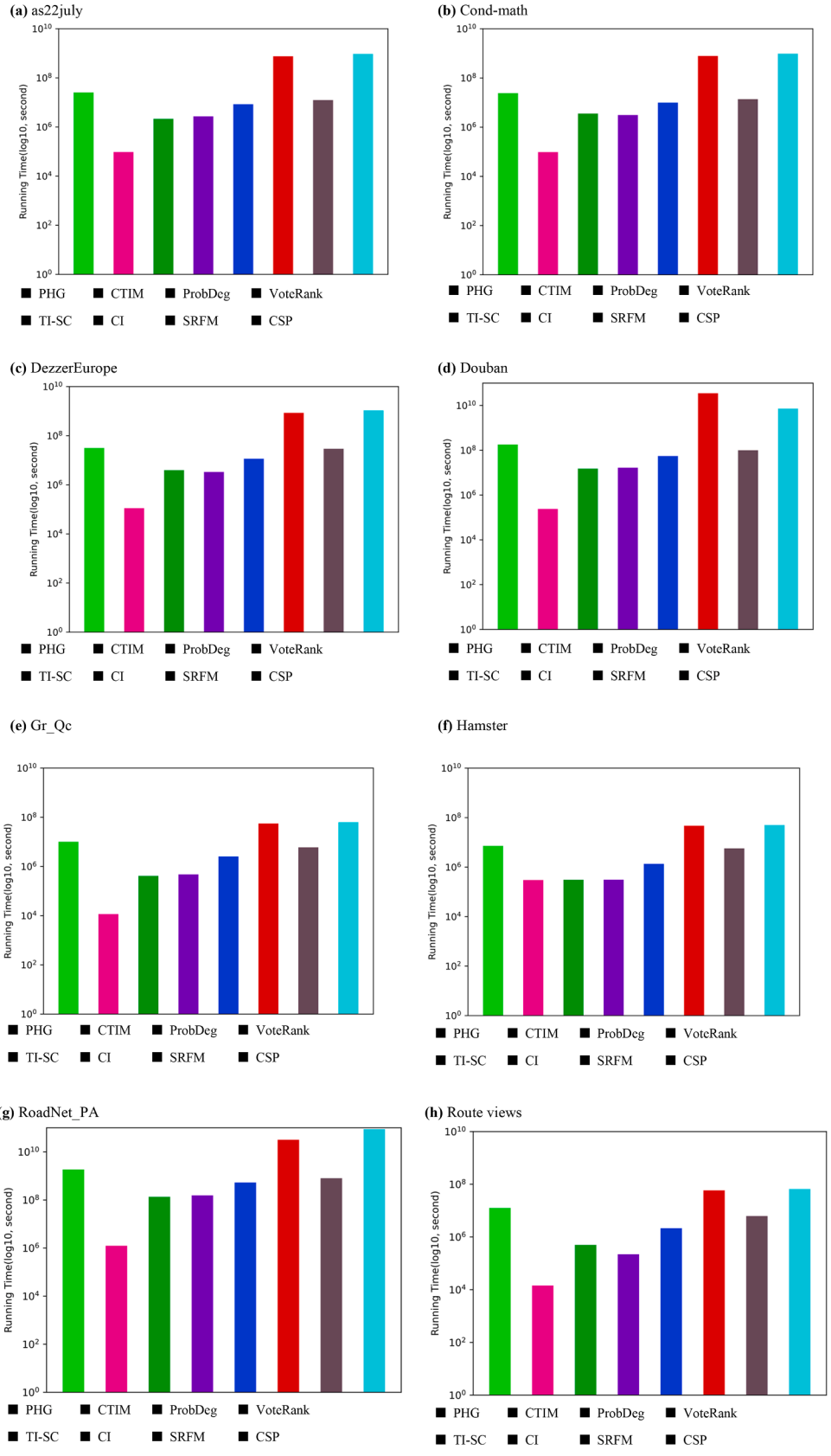
In order to verify the efficiency of the CSP algorithm in searching the most influential nodes in satisfactory running time, we have evaluated the running time of all algorithms on benchmark networks. Fig. 5 shows the running time of all algorithms for the influence probability p = 0.01 and k = 30.

Fig. 5 shows that CSP, VoteRank, and ProbDegree can identify the target seed nodes on all networks in less than 10 s. However, considering the obtained influence spread by these algorithms, VoteRank and ProbDegree algorithms always are less effective than the CSP algorithm. In contrast, Fig. 5 reveals that CI, PHG, and CTIM algorithms are the most time-consuming, and it takes several hours to identify the target seed set on some networks whereas their performance is not superior than some other algorithms.

To evaluate the CSP algorithm's efficiency in identifying influential nodes within a reasonable time, the running times of all algorithms were tested on benchmark networks. Fig. 5 displays the running times of all algorithms for the influence probability p = 0.01 and k = 30. The results reveal that CSP, VoteRank, and ProbDegree can identify the target seed nodes in less than 10 s on all networks. However, VoteRank and ProbDegree algorithms are less effective than the CSP algorithm in terms of the obtained influence spread. On the other hand, CI, PHG, and CTIM algorithms are the most time-consuming, taking several hours to identify the target seed set on some networks, while their performance is not superior to some other algorithms.

6. Conclusion

The study of influence maximization in social network analysis remains a significant research topic. Developing effective, efficient, and scalable methods for this purpose is crucial. One of the main challenges in the influence maximization problem is the large search



(caption on next page)

Fig. 5. Running Time comparison in 8 datasets under the IIC model.

space for identifying key influential seed nodes to maximize information diffusion in a network. To address this issue, we proposed the CSP algorithm, which provides fast module detection, module filtering, node filtering, and candidate node selection in linear time complexity. The CSP algorithm applies to filter actions to both modules and remaining nodes to reduce the search space and selects candidate nodes based on the *limit* criterion. Finally, the best candidate nodes are selected based on their importance and strategic position to form the final seed set. Our experiments on eight networks demonstrated that the CSP algorithm is more effective and accurate in identifying influential nodes with low time complexity compared to other state-of-the-art methods. We also extended the traditional IC model to the IIC model to match with real-world applications.

As future research direction, deep learning algorithms can be used to modify the influence probability and threshold limit in diffusion models to enhance the spread of influence in social networks. Moreover, the optimal approximation is nearly guaranteed in social networks. Hybrid approaches and meta-heuristic algorithms are also potential research areas to choose seed nodes. Additionally, the parallel calculation of influence spread and the selection of seed nodes is one of the potential future research areas to reduce the runtime of influence maximization algorithms and allows for faster seed node selection. Additionally, because influence maximization is an optimization problem, meta-heuristic algorithms are the best option for hybrid approaches.

CRediT authorship contribution statement

Hamid Ahmadi Beni: Conceptualization, Software, Writing - original draft, Validation. **Asgarali Bouyer:** Conceptualization, Methodology, Supervision, Writing - review & editing. **Sevda Azimi:** Software, Resources, Visualization. **Alireza Rouhi:** Investigation, Validation. **Bahman Arasteh:** Data curation, Review & Editing and rewriting.

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

[1] S.P. Borgatti, D.S. Halgin, On network theory, Org. Sci. 22 (5) (2011) 1168–1181.
[2] C. Dong, Y. Li, H. Gong, M. Chen, J. Li, Y. Shen, M. Yang, A survey of natural language generation, ACM Comput. Surv. 55 (8) (2023) 1–38.
[3] X. Liu, et al., What matters in the e-commerce era? Modelling and mapping shop rents in Guangzhou, China, Land Use Policy 123 (2022), 106430.
[4] F. Meng, X. Xiao, J. Wang, Rating the crisis of online public opinion using a multi-level index system, Int. Arab J. Inf. Technol. 19 (4) (2022) 597–608.
[5] X. Qin, Z. Liu, Y. Liu, S. Liu, B.o. Yang, L. Yin, M. Liu, W. Zheng, User OCEAN personality model construction method using a BP neural network, Electronics 11 (19) (2022), 3022.
[6] B. Rozemberczki, O. Kiss, R. Sarkar, Karate Club: an API oriented open-source python framework for unsupervised learning on graphs. Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020.
[7] Q. Ni, et al., Influence-based community partition with sandwich method for social networks, IEEE Trans. Comput. Social Syst. (2022).
[8] A. Bouyer, K. Azad, A. Rouhi, A fast community detection algorithm using a local and multi-level label diffusion method in social networks, Int. J. Gen. Syst. 51 (4) (2022) 352–385.
[9] S. Taheri, A. Bouyer, Community detection in social networks using affinity propagation with adaptive similarity matrix, Big Data 8 (3) (2020) 189–202.
[10] N. Samadi, A. Bouyer, Identifying influential spreaders based on edge ratio and neighborhood diversity measures in complex networks, Computing 101 (8) (2019) 1147–1175.
[11] S. Peng, Y. Zhou, L. Cao, S. Yu, J. Niu, W. Jia, Influence analysis in social networks: A survey, J. Netw. Comput. Appl. 106 (2018) 17–32.
[12] Z. Noshad, A. Bouyer, M. Noshad, Mutual information-based recommender system using autoencoder, Appl. Soft Comput. 109 (2021), 107547.
[13] G. Appel, L. Grewal, R. Hadi, A.T. Stephen, The future of social media in marketing, J. Acad. Mark. Sci. 48 (1) (2020) 79–95.
[14] A. Bouyer, H. Roghani, LSMD: a fast and robust local community detection starting from low degree nodes in social networks, Futur. Gener. Comput. Syst. 113 (2020) 41–57.
[15] M. Mahmoudi, *COVID lessons: was there any way to reduce the negative effect of COVID-19 on the United States economy?* arXiv preprint arXiv:2201.00274, 2022.
[16] Z. Aghaee, M.M. Ghasemi, H.A. Beni, A. Bouyer, A. Fatemi, A survey on meta-heuristic algorithms for the influence maximization problem in the social networks, Computing 103 (11) (2021) 2437–2477.
[17] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.
[18] J. Leskovec, et al., Cost-effective outbreak detection in networks. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007.
[19] R. Narayanam, Y. Narahari, A shapley value-based approach to discover influential nodes in social networks, IEEE Trans. Autom. Sci. Eng. 8 (1) (2010) 130–147.
[20] Q. Liqing, G.u. Chunmei, Z. Shuang, T. Xiangbo, Z. Mingjv, TSIM: A two-stage selection algorithm for influence maximization in social networks, IEEE Access 8 (2020) 12084–12095.
[21] S. Galhotra, et al., Asim: A scalable algorithm for influence maximization under the independent cascade model. Proceedings of the 24th International Conference on World Wide Web, 2015.
[22] H. Zhang, L. Fu, J. Ding, F. Tang, Y. Xiao, X. Wang, G. Chen, C. Zhou, Maximizing the spread of effective information in social networks, IEEE Trans. Knowl. Data Eng. 35 (4) (2023) 4062–4076.
[23] M.M. Keikha, et al., Influence maximization across heterogeneous interconnected networks based on deep learning, Expert Syst. Appl. 140 (2020), 112905.
[24] J. Dai, J. Zhu, G. Wang, Opinion influence maximization problem in online social networks based on group polarization effect, Inf. Sci. 609 (2022) 195–214.
[25] H. Ahmadi Beni, A. Bouyer, Identifying influential nodes using a shell-based ranking and filtering method in social networks, Big Data 9 (3) (2021) 219–232.

[26] A. Bouyer, H.A. Beni, Influence maximization problem by leveraging the local traveling and node labeling method for discovering most influential nodes in social networks, Physica A 592 (2022), 126841.
[27] WeiMin Li, Z. Li, A.M. Luvembe, C. Yang, Influence maximization algorithm based on Gaussian propagation model, Inf. Sci. 568 (2021) 386–402.
[28] G. Rao, D. Li, Y. Wang, W. Chen, C. Zhou, Y. Zhu, Maximizing the influence with κ-grouping constraint, Inf. Sci. 629 (2023) 204–221.
[29] D. Bucur, G. Iacca, Influence maximization in social networks with genetic algorithms. European Conference on the Applications of Evolutionary Computation, Springer, 2016.
[30] J. Lv, J. Guo, H. Ren, Efficient greedy algorithms for influence maximization in social networks, J. Inf. Process. Syst. 10 (3) (2014) 471–482.
[31] C.-W. Tsai, Y.-C. Yang, M.-C. Chiang. *A genetic newgreedy algorithm for influence maximization in social network*, in: *2015 IEEE International Conference on Systems, Man, and Cybernetics*. 2015. IEEE.
[32] T.K. Biswas, A. Abbasi, R.K. Chakrabortty, An MCDM integrated adaptive simulated annealing approach for influence maximization in social networks, Inf. Sci. 556 (2021) 27–48.
[33] Z. Liang, Q. He, H. Du, W. Xu, Targeted influence maximization in competitive social networks, Inf. Sci. 619 (2023) 390–405.
[34] J. Jabari Lotf, M. Abdollahi Azgomi, M.R. Ebrahimi Dishabi, An improved influence maximization method for social networks based on genetic algorithm, Physica A 586 (2022), 126480.
[35] S. Kumar, A. Mallik, A. Khetarpal, B.S. Panda, Influence maximization in social networks using graph embedding and graph neural network, Inf. Sci. 607 (2022) 1617–1636.
[36] L. Cui, H. Hu, S. Yu, Q. Yan, Z. Ming, Z. Wen, N. Lu, DDSE: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks, J. Netw. Comput. Appl. 103 (2018) 119–130.
[37] M. Zarezadeh, E. Nourani, A. Bouyer, DPNLP: distance based peripheral nodes label propagation algorithm for community detection in social networks, World Wide Web 25 (1) (2022) 73–98.
[38] H. Roghani, A. Bouyer, A fast local balanced label diffusion algorithm for community detection in social networks, IEEE Trans. Knowl. Data Eng. (2022).
[39] Y., Wang, et al. *Community-based greedy algorithm for mining top-k influential nodes in mobile social networks*, in: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2010.
[40] H.A. Beni, A. Bouyer, TI-SC: top-k influential nodes selection based on community detection and scoring criteria in social networks, J. Ambient Intell. Hum. Comput. 11 (11) (2020) 4889–4908.
[41] J. Shang, S. Zhou, X. Li, L. Liu, H. Wu, CoFIM: A community-based framework for influence maximization on large-scale networks, Knowl.-Based Syst. 117 (2017) 88–100.
[42] A. Bouyer, et al., FIP: A fast overlapping community-based Influence Maximization Algorithm using probability coefficient of global diffusion in social networks, Expert Syst. Appl. 213 (2023), 118869.
[43] F. Kazemzadeh, A. Asghar Safaei, M. Mirzarezaee, S. Afsharian, H. Kosarirad, Determination of influential nodes based on the Communities' structure to maximize influence in social networks, Neurocomputing 534 (2023) 18–28.
[44] K.V. Rao, C.R. Chowdary, CBIM: Community-based influence maximization in multilayer networks, Inf. Sci. 609 (2022) 578–594.
[45] G. D'Angelo, L. Severini, Y. Velaj, Influence maximization in the independent cascade model. ICTCS, Citeseer, 2016.
[46] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Trans. Knowl. Discov. Data (TKDD) 1 (1) (2007), 2-es.
[47] J. Kunegis, *Konect: the koblenz network collection*, in: *Proceedings of the 22nd international conference on world wide web*. 2013.
[48] R. Zafarani, H. Liu, *Social Computing Data Repository at ASU [*http://socialcomputing. asu. edu*]. Tempe, AZ: Arizona State University, School of Computing.* Informatics and Decision Systems Engineering, 2009.
[49] J. Leskovec, A. Krevl, *SNAP Datasets: Stanford Large Network Dataset Collection*. 2014, Ann Arbor, MI, USA.
[50] R.L. Harrison, Introduction to monte carlo simulation. AIP Conference Proceedings, American Institute of Physics, 2010.