



# An Influence Maximization Algorithm Based on Improved K-Shell in Temporal Social Networks

Wenlong Zhu<sup>1,\*</sup>, Yu Miao<sup>1</sup>, Shuangshuang Yang<sup>2</sup>, Zuozheng Lian<sup>1</sup> and Lianhe Cui<sup>1</sup>

<sup>1</sup>College of Computer and Control Engineering of Qiqihar University, Qiqihar, 161006, China

<sup>2</sup>College of Teacher Education of Qiqihar University, Qiqihar, 161006, China

\*Corresponding Author: Wenlong Zhu. Email: zwl\_qqhr@qqhru.edu.cn

Received: 19 September 2022; Accepted: 06 January 2023

**Abstract:** Influence maximization of temporal social networks (IMT) is a problem that aims to find the most influential set of nodes in the temporal network so that their information can be the most widely spread. To solve the IMT problem, we propose an influence maximization algorithm based on an improved K-shell method, namely improved K-shell in temporal social networks (KT). The algorithm takes into account the global and local structures of temporal social networks. First, to obtain the kernel value  $K_s$  of each node, in the global scope, it layers the network according to the temporal characteristic of nodes by improving the K-shell method. Then, in the local scope, the calculation method of comprehensive degree is proposed to weigh the influence of nodes. Finally, the node with the highest comprehensive degree in each core layer is selected as the seed. However, the seed selection strategy of KT can easily lose some influential nodes. Thus, by optimizing the seed selection strategy, this paper proposes an efficient heuristic algorithm called improved K-shell in temporal social networks for influence maximization (KTIM). According to the hierarchical distribution of cores, the algorithm adds nodes near the central core to the candidate seed set. It then searches for seeds in the candidate seed set according to the comprehensive degree. Experiments show that KTIM is close to the best performing improved method for influence maximization of temporal graph (IMIT) algorithm in terms of effectiveness, but runs at least an order of magnitude faster than it. Therefore, considering the effectiveness and efficiency simultaneously in temporal social networks, the KTIM algorithm works better than other baseline algorithms.

**Keywords:** Temporal social network; influence maximization; improved K-shell; comprehensive degree

## 1 Introduction

In the era of mobile Internet, online social networks have become an important channel for information dissemination and have greatly impacted the lives of people. Among the many studies

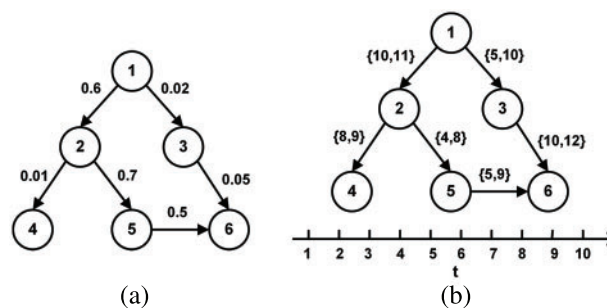


This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

on social networks, influence maximization (IM) is one of the most popular research directions which is first proposed by Domingos et al. [1]. IM aims to find a set of nodes to spread influence as much as possible within the network. Nowadays, this kind of research has been widely used in viral marketing. For example, a company will select the most influential users in the hope of creating a word-of-mouth effect for a series of products.

To solve the IM problem, most researchers usually focus on the topology of traditional social networks and abstract the social network into a static graph, ignoring the temporal nature of the network. However, in real social networks, such as telephone and mail transmission networks, nodes are not always connected. Instead, they are connected at a specific time, and the connections between nodes are time-series, which means that the networks are temporal. Therefore, in these temporal social networks, it is not enough to determine the closeness of users by their connections with each other. It is also necessary to record more specific interaction times between users to accurately portray their relationships. In this way, we can identify nodes that play a critical role in the influence propagation and effectively address the issue of IM.

As shown in Fig. 1, it illustrates the difference between traditional and temporal social networks in computing the influence of nodes in the IM problem. Fig. 1a shows a traditional social network  $G$ , and the weights on the edges indicate the influence propagation probability of nodes. Fig. 1b represents a temporal social network  $G_T$  with the same probability of influence propagation between nodes as in  $G$ , and the weights on the edges indicate the connections between nodes at a particular time. Next, the influence of node 1 in these two networks is calculated respectively. In  $G$ , node 1 attempts to activate its neighbor nodes 2 and 3. If only node 2 is successfully activated, node 2 becomes active at this time, so it can try to activate its neighbor nodes 4 and 5. If only node 5 is successfully activated, node 5 attempts to activate its neighbor node 6. If the activation fails, then node 1 can affect two nodes. So the influence of node 1 is 2 in  $G$ . But in  $G_T$ , assuming that node 1 also activates node 2, and the weights on the edge between nodes 1 and 2 indicate that these two nodes are only connected at times 10 and 11, which means that node 2 is activated at time 10 at the earliest, i.e., node 2 is inactive before time 10. However, node 2 and its neighbor nodes 4 and 5 are connected only before time 9, so node 2 cannot activate its neighbors. Thus, the influence of node 1 is 1 in  $G_T$ .



**Figure 1:** Two examples of social networks. (a) Shows a traditional social network. (b) Shows a temporal social network

As mentioned above, due to the temporal nature of social networks, the topology between nodes changes dynamically over time, which makes traditional influence propagation models and influence maximization algorithms unsuitable for temporal social networks. In the past few years, some studies [2,3] have solved the IMT problem by using greedy algorithms and optimizing the computation of marginal gain, but these algorithms are time-consuming and unsuitable for temporal social networks.

Some studies [4,5] have modeled temporal networks by snapshots to solve the IMT problem, but these methods are too cumbersome when the network size is large. However, Kitsak et al. [6] find that the number of cores has more stable propagation than node attributes such as degree and betweenness centrality. They propose the K-shell algorithm, which can determine the significance of different node positions in a network and quickly identify the most influential nodes. This concept applies to temporal social networks.

Therefore, this paper uses the temporal graph to model the temporal social network from a global perspective and proposes an improved K-shell method for effectively solving the IMT problem, namely the KT algorithm. Using the static and dynamic attributes of nodes, the algorithm distinguishes the influence of nodes from the global and local structure of the temporal social network. First, in the global scope of the network, an improved K-shell method is proposed, which stratifies the network based on the temporal characteristic of nodes and obtains the kernel value  $K_s$  of each node. The  $K_s$  value represents the static properties of nodes. Further, in the local scope of each core layer, a calculation method of comprehensive degree is proposed, which weighs the direct and indirect influence of nodes. The comprehensive degree represents the dynamic properties of nodes. On this basis, the KT algorithm is proposed, which selects the node with the highest comprehensive degree as the seed node at each core layer from the central core to the edge core. However, the seed selection strategy of KT tends to lose some influential nodes, resulting in a decrease in the influence of the seeds. Therefore, we optimize the seed selection strategy and propose the KTIM algorithm, which adds the nodes near the central core to the candidate seed set, and then selects the top  $k$  nodes with the highest comprehensive degree in the set as the seed nodes. It has low time complexity and is similar to the effectiveness of the baseline algorithm, which can effectively solve the IMT problem. In summary, our main contributions to this paper are as follows:

- 1) We define the problem of IMT and provide an idea to solve it by considering the global and local structure of the network.
- 2) Based on the above idea, we propose the KT algorithm to solve the IMT problem. The algorithm stratifies the network based on an improved K-shell method and then selects the optimal seed according to the comprehensive degree of nodes.
- 3) We further propose a more effective KTIM algorithm by optimizing the seed selection strategy to solve the IMT problem.
- 4) Through experiments on four real-world temporal datasets, we demonstrate the efficiency and effectiveness of our proposed algorithms.

In the rest of this paper, Section 2 introduces related works. We formulate the IMT problem and propose a greedy based algorithm to solve it in Section 3. The fourth section describes the KT and KTIM algorithms. Section 5 provides experimental results on real-world datasets, and we conclude this paper in Section 6.

## 2 Related Work

In this section, we briefly review the related work on solutions to the IM problem, including influence maximization algorithms in traditional and temporal social networks.

### 2.1 Influence Maximization Algorithms in Traditional Social Networks

In traditional social networks, influence maximization algorithms mainly include greedy and heuristic algorithms. Kempe et al. [7] first formulate the IM problem as a discrete optimization

problem of finding the  $k$  nodes with maximum influence on a particular influence propagation model and prove that the IM problem is NP-hard. On this basis, they propose a greedy algorithm by utilizing Monte Carlo simulations and prove that the optimal result can reach 63% of the approximate optimum, but the time complexity of the greedy algorithm is high and unsuitable for large-scale social networks. Further, Leskovec et al. [8] propose the cost-effective lazy forward (CELF) algorithm, which optimizes the traditional greedy algorithm by using the submodel and monotonic properties. Goyal et al. [9] devise the CELF++ algorithm, which further optimizes the CELF algorithm and improves its effectiveness. Experiments show that the CELF++ algorithm is 35%–55% faster than the CELF algorithm. In addition, Chen et al. [10] propose two improved greedy algorithms called NewGreedy and MixedGreedy to further optimize the greedy algorithm by constructing subgraphs of the network. However, the complexity of the above improved greedy algorithms is still high, thus making these algorithms unsuitable for large-scale social networks.

To overcome the inefficiency of the greedy algorithm, researchers have proposed a large number of heuristic algorithms. The issues [11–13] utilize the reverse influence sampling (RIS) method to solve the IM problem, avoiding the limitations of greedy algorithms. The RIS algorithm generates a certain number of random reverse reachable sets (RRsets) through a static graph, and then sequentially selects the node that covers the most RRsets as the seed node until  $k$  seed nodes are selected. In addition, Chen et al. [14] propose the prefix excluding maximum influence arborescence (PMIA) algorithm, which speeds up the calculation of influence propagation probabilities of nodes by constructing the local tree structure of each node. This algorithm has a stable influence range and runs three orders of magnitude faster than the greedy algorithm. However, the PMIA algorithm requires a large amount of memory. Jung et al. [15] propose the influence ranking influence estimation (IRIE) algorithm, which first calculates the influence ranking (IR) of each node through the global influence ranking algorithm and selects the influential nodes according to the ranking order. To avoid the overlapping problem of node influence, after selecting a seed node, the influence estimation (IE) algorithm is used to estimate the influence change of the remaining nodes in the network. Then the influence ranking of the node is updated according to the influence changes of its related nodes. This cycle repeats until  $k$  influential nodes are found. Chen et al. [10] propose the DegreeDiscount algorithm to solve the influence overlap problem of traditional degree estimation algorithms. The algorithm is based on the idea that when a node  $v$ 's neighbor node is selected as a seed node, its degree is discounted when calculating its degree. This algorithm is superior to the simple degree algorithm [16]. Cao et al. [17] propose a core covering algorithm (CCA) based on the hierarchical features of social networks and introduce a coverage distance parameter to avoid the overlap of the influence of nodes.

In recent years, many researchers have extended their research on the IM problem of social networks. Wang et al. [18] discuss the robustness of the influence maximization problem against link attacks and try to find a robust set of seed nodes from the network to deal with the challenge of structural damage to influence propagation. Wu et al. [19] study the IM problem with time constraints, i.e., how to maximize influence spread within a given time range. They propose a time-constrained method based on the topological attributes of nodes and the time attributes of social activities. Wang et al. [20] focus on the trust relationship between nodes in competitive social networks and study the influence problem by simulating the positive and negative propagation of trust nodes. Li et al. [21] focus on the geographical location of individuals and propose a location-oriented IM problem.

## 2.2 Influence Maximization Algorithms in Temporal Social Networks

In temporal social networks, the aim of maximizing influence is to find the most influential seed set under the temporal relationship so that the information propagation of this set is the most. Therefore, it is important to focus not only on topological features but also on temporal features [22,23].

Recently, Wu et al. [2] abstract temporal social networks into temporal graphs, improve the traditional independent cascade model, and propose an independent cascade model on temporal graph (ICT) model. On this basis, they propose the IMIT algorithm to solve the IM problem in temporal social networks by using Monte Carlo simulations. However, the algorithm is unsuitable for temporal social networks because it uses a greedy algorithm that takes a long time. Chen et al. [3] propose the two-stage impact maximization (TIM) algorithm, which uses node characteristics to select candidate seeds in the temporal heuristic stage, and optimizes the marginal gains of nodes for seed selection in the temporal greedy stage. Compared with the IMIT algorithm, the running time of TIM is reduced. However, the temporal greedy stage still takes a long time, which is unsuitable for temporal social networks.

In addition, Zhang et al. [4] propose a prediction and replacement based influence maximization framework using machine learning techniques, which first predicts upcoming network snapshots using historical network snapshot information, and then mines seed nodes suitable for dynamic networks based on the prediction results. Chandran et al. [5] study temporal influence maximization on dynamic social networks and proposes a dynamic influence based seed selection method which estimates the influence of each node by introducing a two-hop triangular influence.

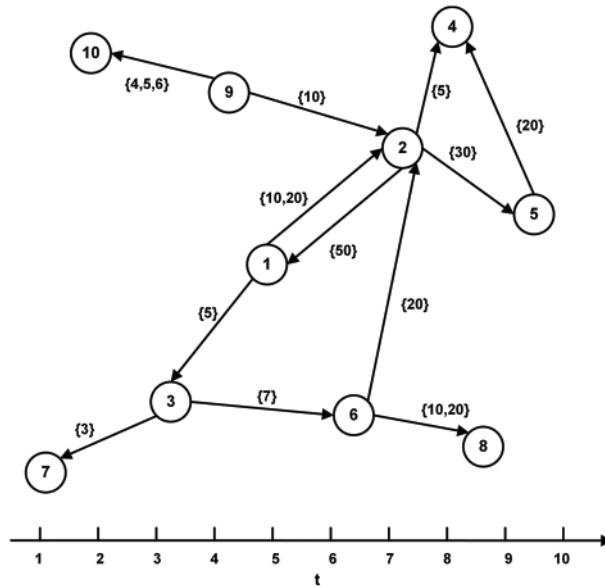
In conclusion, although the influence of the heuristic algorithm is unstable under different social networks and propagation models, its performance can be improved by several orders of magnitude compared to that of the greedy algorithm. This is due to the fact that the greedy algorithm obtains the largest influence nodes using time-consuming Monte Carlo simulations under a particular influence propagation model. In contrast, the heuristic algorithm is not constrained by the influence propagation model and selects the most influential nodes based on the characteristics of nodes and edges. Although the influence propagation effect of the greedy algorithm is excellent and stable, its time complexity is high and unsuitable for temporal social networks.

## 3 Preliminaries

### 3.1 Basic Definition

**Definition 1:** (Temporal social network). Given a network  $G_T(V, E, T_E)$  represents a social network graph with temporal relationships between nodes,  $V$  represents the set of nodes,  $E$  represents the set of edges, where  $|V| = n$ ,  $|E| = m$  and  $T_E$  represents the set of connection times between nodes.  $T_{(u,v)}$  represents the set of times when there is a connection between nodes  $u$  and  $v$ , and  $T_{(u,v)} \in T_E$ .

In a temporal social network, nodes are connected only at a specific time. As shown in Fig. 2, the weights on the edges indicate the time when a connection exists between two nodes. E.g.,  $T_{(1,2)} = \{10, 20\}$  indicates that there is a connection between node 1 and node 2 at times 10 and 20, and there is no connection at other times.



**Figure 2:** Temporal social network  $G_T$

### 3.2 Influence Propagation Probability

**Definition 2:** (Influence propagation probability). The probability that an inactive node  $v$  is successfully influenced by its active parent node  $u$  through a directed edge  $(u, v)$  is the influence propagation probability, denoted as  $P_{u,v} \in [0, 1]$ .

In traditional social networks, degree estimation is usually used to calculate the influence propagation probability between nodes. However, it is not appropriate in temporal social networks, as mentioned in Fig. 1. Due to the addition of temporal relationships, the weight of each edge is related to the number of connections between two nodes, implying that the larger the value of  $|T_{(u,v)}|$ , the larger the value of weight. Therefore, the calculation of the influence propagation probability is based on the method proposed by Chen et al. [3], which is calculated as follows:

$$P_{u,v} = \frac{|T_{(u,v)}|}{\sum_{v \in In(v)} |T_{(u,v)}|} \quad (1)$$

where  $|T_{(u,v)}|$  is the number of connections between nodes  $u$  and  $v$ , and  $In(v)$  represents all in-degree nodes of  $v$ .

### 3.3 Influence Propagation Model

**Definition 3:** (Node active start time). The moment when node  $v$  is successfully activated by its active parent node  $u$  is its active start time, denoted as  $Act_v = \min\{t | t \in T_{(u,v)} \text{ and } t \geq Act_u\}$ .

In traditional social networks, it is not necessary to consider the start time of a node being activated for the IM problem. In contrast, the start time of the node being successfully activated in the temporal social network needs to be considered. Therefore, in this paper, we adopt the ICT model [2] to simulate the influence propagation of nodes. In the initial network, the active start time of all nodes  $v$  can be recorded as  $Act_v = -1$ , indicating that all nodes are inactive. When a seed node  $u$  is selected, we formulate the information diffusion process as follows:

- 1) Let the initial active start time of the seed node be  $Act_u = 0$ , that is, the seed node is active at time 0. At this time, the seed node  $u$  activates its inactive neighbor node  $v$  with a certain probability, and node  $u$  has one and only one chance to try to activate node  $v$ .
- 2) When node  $u$  tries to activate node  $v$ , it judges whether the condition  $Act_u \leq \max(T(u, v))$  is satisfied. If not, it skips node  $v$  and starts trying to activate the next neighbor node. Otherwise, node  $u$  activates node  $v$  with a probability  $P_{u,v}$ .
- 3) Regardless of whether node  $u$  can activate node  $v$ , node  $u$  does not try to activate node  $v$  again in the subsequent propagation process.
- 4) If node  $v$  is successfully activated, its initial active time  $Act_v$  is recorded, where  $Act_v \in T_{(u,v)}$ ,  $Act_u \leq Act_v \leq \max(T_{(u,v)})$ .
- 5) The influence diffusion process is from new active nodes to their inactive neighbors until no new nodes are activated in the network.

### 3.4 The Definition of the IMT Problem

According to the above description, this section introduces the concepts of node influence and marginal gain, and then defines the IMT problem.

**Definition 4:** (Node influence). Node influence is the number of nodes in the network that can be successfully activated by node  $u$ , denoted as  $\sigma(u)$ . Similarly,  $\sigma(S)$  indicates how many nodes can be activated by the seed set  $S$ .

**Definition 5:** (Marginal gain). The marginal gain of a node  $u$  refers to the increment revenue  $\sigma_u(S)$  from adding node  $u$  to the seed set  $S$ .

$$\sigma_u(S) = \sigma(S \cup \{u\}) - \sigma(S) \quad (2)$$

**Definition 6:** (The influence maximization of temporal social networks problem). Given a temporal social network  $G_T(V, E, T_E)$  and a specific influence propagation model such as ICT, the influence maximization of temporal social networks (IMT) problem selects  $k$  nodes as the seed set  $S$  so that the influence range  $\sigma(S)$  generated by  $S$  is the largest. That is, compute  $S^*$  such that:

$$S^* = \operatorname{argmax}_{S \subseteq V, |S|=k} \sigma(S) \quad (3)$$

where  $V$  are nodes set in the network,  $k$  is the size of the seed set,  $\sigma(S)$  represents the influence range of the seed set  $S$ .

### 3.5 Traditional Solution to the IMT Problem

For the IMT problem, the traditional solution follows the idea of the greedy based algorithm, which mainly divides the problem into two sub-processes. One process is calculating the marginal gain of a single node joining the seed set by Monte Carlo simulation approximation. The other process is selecting the  $k$  most influential nodes using the greedy approximation algorithm.

Let  $G_T(V, E, T_E)$  represent the temporal social network,  $k$  is the required number of seed nodes, and  $S$  is the set of seed nodes. We give the pseudocode of the Greedy algorithm as Algorithm 1.

---

#### Algorithm 1: Greedy

---

Input:  $G_T(V, E, T_E), k$

Output:  $S$

1: Initialize  $S \leftarrow \emptyset$

2: while  $(|S| < k)$

---

(Continued)



**Algorithm 1:** Continued

---

```

3:   for each node  $u$  ( $u \in G_T, u \notin S$ )
4:      $\sigma_u(S) = \sigma(S \cup \{u\}) - \sigma(S)$ 
5:   end for
6:    $u \leftarrow \operatorname{argmax}_{u \in V \setminus S} \sigma_u(S)$ 
7:    $S \leftarrow S \cup \{u\}$ 
8: end while
9: return  $S$ 

```

---

Let  $n = |V|$ ,  $m = |E|$ , and  $R$  be simulation times. Since the algorithm runs  $k$  iterations, in each iteration, it needs to go through all nodes  $u \in V \setminus S$  to find the one with the maximum marginal influence. For each node, the algorithm needs to run  $R$  simulations to calculate the marginal gain of the node after joining it into the seed set. Each simulation may need to traverse all edges, so the time complexity of Algorithm 1 is  $O(knRm)$ . As the size of the network increases, the Greedy algorithm becomes very complex in terms of time complexity, making it unsuitable for temporal social networks. So, in the next section, we'll talk about two better algorithms to solve the IMT problem based on an improved k-shell method.

#### 4 Influence Maximization Algorithms of Temporal Social Networks

Although the Greedy algorithm can solve the IMT problem, it is time-consuming. In this section, we propose KT and KTIM algorithms to solve it. Fig. 3 shows the architecture diagram of our work.

As shown in Fig. 3, given a temporal social network  $G_T$ , we first calculate the temporal characteristic of nodes, denoted as  $T(v)$ . Then we propose an improved K-shell method by exploiting the temporal characteristic of nodes to stratify the network and obtain the  $Ks$  value  $V(Ks)$  of each node. Then, we propose the KT algorithm by designing the seed selection strategy. More precisely, the KT algorithm calculates the comprehensive degree of nodes at each core layer, then selects the node with the highest comprehensive degree as the seed node from the central core to the edge core. Subsequently, we propose the KTIM algorithm by optimizing the seed selection strategy. KTIM adds the nodes near the central core to the candidate seed set, and then selects the top  $k$  nodes with the highest comprehensive degree in the candidate seed set as the seed nodes. Finally, we estimate the influence spread of the seed nodes using the influence propagation model.

##### 4.1 Improved K-shell Method

The traditional K-shell algorithm is used to stratify the network based on the degree of nodes in the global scope. Firstly, it continuously removes nodes of degree 1 and their connected edges from the network  $G$  until such nodes no longer appear in  $G$ . Then, all the removed nodes are classified as 1-shell layer, and a  $Ks$  value of 1 is assigned to these nodes. Next, it continuously deletes nodes with degree 2 and their connected edges, sets the  $Ks$  value of these nodes to 2, and classifies them as a 2-shell layer. Finally, the algorithm repeats the above process until all nodes in the network are layered and assigned  $Ks$  values.

It should be noted that the layer of nodes with a larger  $Ks$  value is closer to the core layer of the network, and the nodes in the core layer of the network usually have greater influence. The K-shell algorithm reveals the hierarchy of the network structure and can better estimate the influence of nodes in traditional social networks. However, in temporal social networks, the influence of a node is not only affected by its degree but also related to its temporal characteristic.



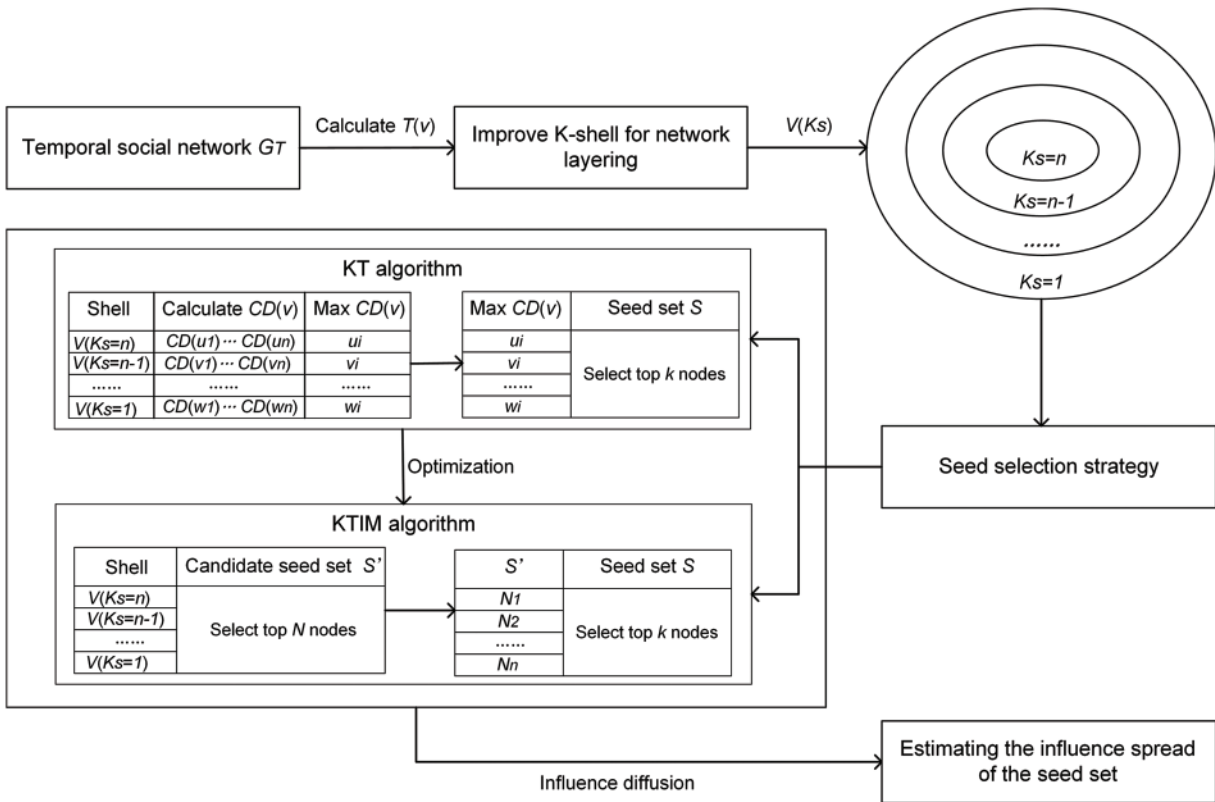


Figure 3: Architecture diagram of our work

As shown in Figs. 4, 4a represents a traditional social network  $G$ . Fig. 4b represents a temporal social network  $G_T$ . In  $G$ , the out-degree of node  $a$  is 2. In  $G_T$ , the out-degree of node  $d$  is also 2, while the number of connections between node  $d$  and its neighbor nodes is 6, which is much larger than the number of connections between node  $a$  and its neighbors. That is, the probability that node  $d$  activates its neighbor nodes is greater than that of node  $a$ . Although the degrees of node  $a$  and node  $d$  are the same, the influence of node  $d$  is greater than that of node  $a$ . However, the traditional K-shell method considers that both nodes have the same influence, so it is unsuitable for temporal social networks.

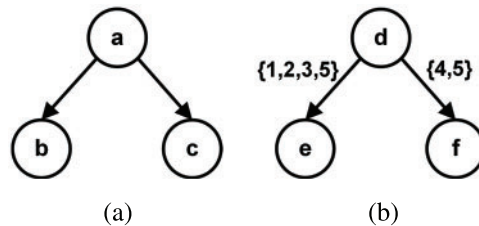


Figure 4: The comparison of node influence. (a) Is a traditional social network  $G$ . (b) Is a temporal social network  $G_T$

As mentioned above, to make the K-shell method applicable to temporal social networks, it is necessary to improve the K-shell method to consider the temporal characteristic of nodes to stratify the network. The temporal characteristic of a node is defined as follows:

$$T(u) = \sum_{v \in Out(u)} |T_{(u,v)}| \tag{4}$$

where  $T(u)$  represents the temporal characteristic of node  $u$ ,  $Out(u)$  represents the outgoing neighbors set of node  $u$ , and  $T_{(u,v)}$  represents the connection times between nodes  $u$  and  $v$ .

Furthermore, we propose an improved K-shell method by exploiting the temporal characteristic of nodes in temporal social networks. The main idea is as follows: First, it continuously removes nodes and their edges whose temporal characteristic value is 1 until such nodes no longer appear. At this time, all deleted nodes are classified as a 1-shell layer and valued to 1. Then, it continuously deletes nodes and their edges with a temporal characteristic value of 2, classifies the deleted nodes as a 2-shell layer, and values these nodes to 2. The above process is repeated until all nodes in the network are classified, and  $K_s$  values are assigned. Among them, nodes in the layer with a larger  $K_s$  value are closer to the core layer of the network, and nodes in the core layer usually have greater influence. We show an example in Fig. 5.

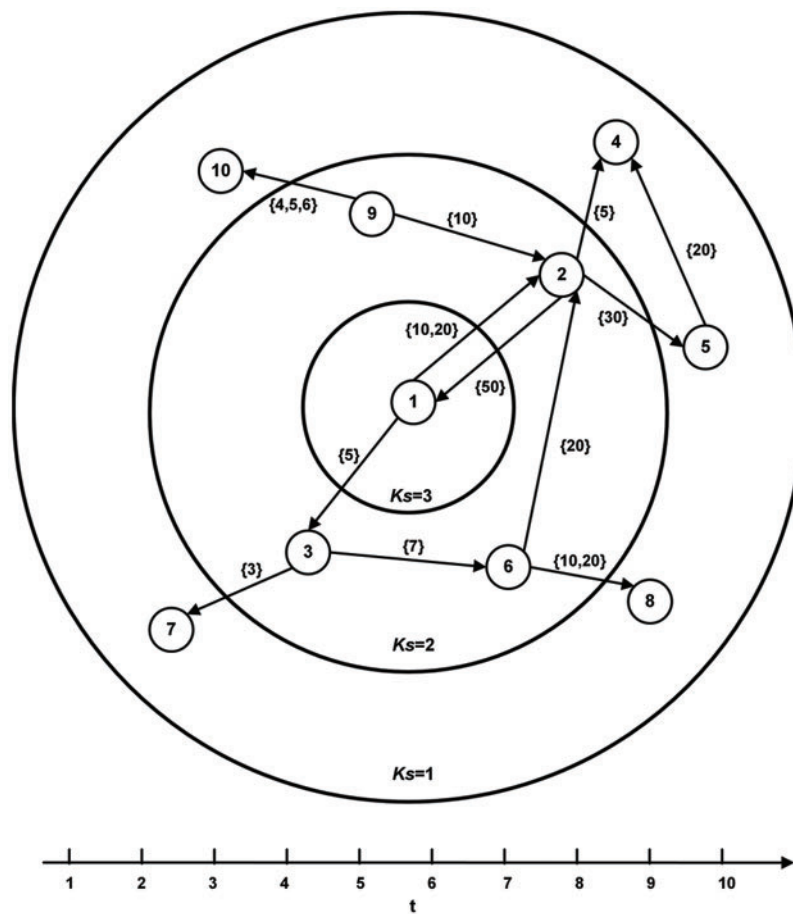


Figure 5: Applying improved K-shell methods for layering a temporal social network

As shown in Fig. 5, it is the result of layering the temporal social network in Fig. 2. The network is stratified into three layers, and the  $Ks$  value gradually increases from the border layer to the core layer, which means that the importance of nodes gradually increases. More precisely, Node 1 belongs to the 3-shell layer and usually has the greatest influence. Nodes 2, 3, 6 and 9 belong to the 2-shell layer and have the same  $Ks$  value. Node 10 and other nodes belong to the 1-shell layer with the same  $Ks$  value and usually have the least influence. It can be seen that the improved K-shell method can provide a more reasonable layering of the temporal social network.

#### 4.2 Seed Selection Strategy

In a temporal social network, after layering the network through the improved K-shell method in Section 4.1, each node is assigned a  $Ks$  value to quantify the importance of the node, representing the static property of the node. But this method is a coarse-grained decomposition method based on the global structure of the network, which makes nodes with high  $Ks$  values easy to cluster together. Therefore, if the seeds are only selected according to the improved K-shell method, it will cause two problems. On the one hand, it is difficult to distinguish the importance of nodes with the same  $Ks$  value in the same kernel layer. As shown in Fig. 5, in the 2-shell layer of the temporal social network, nodes 9, 2, 6, and 3 are in the same layer and have the same  $Ks$ , so it is not easy to distinguish the importance of these nodes. On the other hand, determining the importance of a node according to the value of the static attribute  $Ks$  alone is very likely to cause the selected seed node to be inaccurate.

Therefore, to address the above issues, we further focus on the other characteristics of nodes within the local scope of each kernel layer, so the comprehensive degree is proposed to represent the dynamic properties of nodes and to weigh the direct and indirect effects of nodes in the propagation process. The comprehensive degree is derived from common sense that if a node has more neighbors, and its neighbors also have more neighbors, then that node may have more influence. So, the comprehensive degree of a node is defined as follows:

$$CD(u) = out\_degree(u) + \frac{\sum_{v \in Out(u)} |out\_degree(v)|}{|out\_degree(u)|} \quad (5)$$

where  $CD(u)$  represents the comprehensive degree of node  $u$ ,  $out\_degree(u)$  represents the out-degree of node  $u$ , and  $Out(u)$  represents the set of out neighbors of node  $u$ .

As shown in Fig. 5, the comprehensive degree of each node in the second layer of the temporal social network is calculated as  $CD(2) = 3 + \frac{3}{3} = 4.0$ ,  $CD(9) = 2 + \frac{3}{2} = 3.5$ ,  $CD(6) = 2 + \frac{3}{2} = 3.5$ ,  $CD(3) = 2 + \frac{2}{2} = 3.0$ . It can be seen that node 2 has the greatest influence in the 2-shell layer network. Therefore, with the above solution, the nodes have the same static attribute  $Ks$  value in the same kernel layer. However, the dynamic attribute comprehensive degree of the nodes is different, and it can be seen that node 2 with the largest comprehensive degree has the largest influence.

#### 4.3 The KT Algorithm

Based on the above analysis, the KT algorithm is proposed in this paper. The basic idea is that in a temporal social network, the network is first layered according to the temporal characteristic of nodes. Then starting from the core layer of the network, each layer selects the node with the largest comprehensive degree to join the seed set. Let  $G_T(V, E, T_E)$  represent the temporal social network,  $k$  is the required number of seed nodes, and  $S$  is the set of seed nodes. The description of KT is given in Algorithm 2.

**Algorithm 2: KT**


---

Input:  $G_T(V, E, T_E), k$   
Output:  $S$

- 1: Initialize  $S \leftarrow \emptyset$
- 2: for  $u$  in  $G_T$  do
- 3:   calculation  $T(u)$
- 4: end for
- 5:  $G'_T \xleftarrow{T(u)} G_T$
- 6: for  $i = 1$  to  $k$
- 7:   for *shell layer* in *reversed*( $G'_T$ )
- 8:     for  $u$  in shell
- 9:       calculation  $CD(u)$
- 10:    end for
- 11:    $v = \operatorname{argmax}_u \{CD(u) | u \in G'_T/S\}$
- 12:    $S = S \cup \{v\}$
- 13:   end for
- 14: end for

---

The algorithm first initializes the candidate seed set  $S$  (line 1), calculates the temporal characteristic for each node in the graph (lines 2–4), and applies the improved K-shell method to stratify the temporal social network  $G_T$  to generate the layered network  $G'_T$  (line 5). Lines 6–14 iteratively search for the seeds with the highest comprehensive degree in each layer. In each iteration, line 7 indicates the seeds are selected from the core layer to the border layer, and line 12 adds the node with the highest comprehensive degree in the current layer to the seed set  $S$ .

Let the number of nodes be  $n$ , the number of edges be  $m$ , the seed set size be  $k$ , and the number of connections between nodes be  $T$ . For the time complexity, line 1 is a constant time, lines 2–3 take  $O(T)$ , layering the network takes  $O(n)$  in line 5, and lines 6–14 select the node with the largest comprehensive degree that takes  $O(kn)$ . Therefore, the overall time complexity of the KT algorithm is  $O(T + kn)$ .

#### 4.4 The KTIM Algorithm

The KT algorithm distinguishes the importance of nodes with the same  $K_s$  value and selects a node with the maximum comprehensive degree as a seed node in each core layer. However, this selection strategy tends to lose some more important nodes, resulting in inefficient solutions to the IMT problem. Therefore, we propose the KTIM algorithm by optimizing the seed selection strategy.

##### 4.4.1 Optimizing Seeds Selection Strategy

In a temporal social network, the KT algorithm selects the node with the highest comprehensive degree as the seed node for each network layer. However, typically each kernel layer may aggregate multiple nodes. When the KT algorithm selects only one node as a seed node at each kernel layer, other potentially more influential nodes will be lost, which may lead to a reduction in the influence range of the seed set.

As shown in Fig. 5, the comprehensive degree of multiple nodes in the 2-shell layer is greater than that of the nodes in the 1-shell layer. E.g., in the 2-shell layer,  $CD(2) = 3 + \frac{3}{3} = 4.0$ ,  $CD(9) =$

$2 + \frac{3}{2} = 3.5$ , but in the 1-shell layer  $CD(5) = 1 + \frac{0}{1} = 1.0$ . It can be seen that the seed selection strategy of the KT algorithm only selects one node with the highest comprehensive degree as the seed node in each core layer, resulting in ignoring other nodes that may be more influential. Therefore, the seed nodes selected by the seed selection strategy of the KT algorithm may not be the more influential nodes in the whole network and cause a decrease in the influence of the seed set.

Therefore, to address the shortcomings of the seed selection strategy of the KT algorithm, we further optimize the seeds selection strategy by combining the static attribute  $Ks$  value and the dynamic attribute comprehensive degree of nodes to select the seed nodes. The optimized seed selection strategy is mainly divided into two steps, which are described as follows:

1) For the layered network  $G_T(V, E, T_E)$  obtained by the improved K-shell method mentioned in section 4.1, the top  $N$  nodes with higher  $Ks$  values are selected to join the candidate seed set  $S'$ , representing the set of nodes that may become seed nodes. i.e.,

$$S' = \{u | u = [u_i | u_i \in G'_T, Ks(u_i) > Ks', i \in \{1, \dots, n\}]\} \quad (6)$$

where  $u_i$  is a node in  $G'_T$ ,  $Ks(u)$  denotes the static attribute  $Ks$  value of a node  $u$ , and  $Ks'$  is the  $N + 1$ th higher  $Ks$  value.

2) The top  $k$  nodes with a greater comprehensive degree are selected from the candidate seed set  $S'$  and added to the seed set  $S$ , which represents the set of seed nodes. i.e.,

$$S = \{v | v = [v_i | v_i \in S', CD(v_i) > cd', i \in \{1, \dots, N\}]\} \quad (7)$$

where  $v_i$  is a node in  $S'$ ,  $CD(v)$  denotes the dynamic attribute comprehensive degree of a node  $v$ , and  $cd'$  is the  $k + 1$ th higher comprehensive degree.

#### 4.4.2 Execution Process of KTIM

Let  $G_T(V, E, T_E)$  represent the temporal social network,  $k$  is the required number of seed nodes,  $S'$  is the set of candidate seed nodes, and  $S$  is the set of seed nodes. Algorithm 3 shows the pseudocode of the KTIM algorithm.

---

#### Algorithm 3: KTIM

---

Input:  $G_T(V, E, T_E), k$

Output:  $S$

```

1: Initialize  $S \leftarrow \emptyset, S' \leftarrow \emptyset$ 
2: for  $u$  in  $G_T$  do
3:   calculation  $T(u)$ 
4: end for
5:  $G'_T \xleftarrow{T(u)} G_T$ 
6: for shell layer in reversed ( $G'_T$ )
7:   for  $i = 1$  to  $N$  do
8:      $v = \operatorname{argmax}_u \{Ks(u) | u \in G'_T/S'\}$ 
9:      $S' = S' \cup \{v\}$ 
10:  end for
11: end for
12: for  $i = 1$  to  $k$ 

```

---

(Continued)

**Algorithm 3:** Continued

---

```

13: for  $u$  in  $S'$ 
14:     calculation  $CD(u)$ 
15: end for
16:  $v = \operatorname{argmax}_u\{CD(u)|u \in S'/S\}$ 
17:  $S = S \cup \{v\}$ 
18: end for

```

---

In Algorithm 3, line 1 initializes the candidate seed set  $S'$  and seed set  $S$ . Lines 2–4 calculate the temporal characteristic of each node. Line 5 applies the improved K-shell method to stratify the temporal network. Lines 6–11 iteratively find the top  $N$  ( $N$  is the size of the candidate seed set, and  $k < N < n$ ) nodes with larger  $Ks$  value and add them to the candidate seed set  $S'$ . Line 6 means the candidate seeds are selected from the core layer to the border layer. Lines 12–18 execute  $k$  rounds to find  $k$  seeds with the highest comprehensive degree in the candidate seed set  $S'$  and add them to the seed set  $S$ .

Let the number of nodes in the network  $G_T(V, E, T_E)$  be  $n$ , the number of edges be  $m$ , the size of the seed set be  $k$ , and the number of connections between nodes be  $T$ . The time complexity of the KTIM algorithm is calculated as follows: It takes  $O(T)$  to calculate the temporal characteristic of nodes in lines 2–3, and the time complexity of layering the network in line 5 is  $O(n)$ ; Lines 6–11 take  $O(N)$  to find the top  $N$  candidate seeds; Executing  $k$  rounds in line 12 takes  $O(k)$ , and  $O(N)$  to select the current seed in lines 13–17 in each round. Therefore, the time complexity of the KTIM algorithm is  $O(T + n + kN)$ .

## 5 Experiment and Evaluation

We conduct experiments on our proposed algorithms and a series of other compared algorithms on four real-world temporal social network datasets. These experiments aim to evaluate the effectiveness and efficiency of our proposed KT and KTIM algorithms for the IMT problem.

### 5.1 Experimental Setup

**Datasets.** The experiments in this paper are conducted on four real-world temporal network datasets. Each row of data in these databases consists of three items: denoted as  $(u, v, t)$ , which means that  $u$  contacts  $v$  at time  $t$ . the details of the datasets are shown in [Table 1](#).

**Table 1:** Parameters of experimental datasets

Name	Nodes	Temporal edges	Static edges	Time span/day
Email-Eu-core	986	332334	24929	803
CollegeMsg	1899	59835	20296	193
Math overflow	21688	107581	90489	2350
Ask ubuntu	75555	356822	178210	2418

Among them, dataset 1 [24] is generated from the email data of large European research institutions. Dataset 2 [25] consists of private messages sent on the online social network of the

University of California, Irvine. Dataset 3 [26] was generated from a famous website Math Overflow. Dataset 4 [24] originates from a question-and-answer website Ask Ubuntu.

**Evaluated algorithms.** We conduct experimental comparisons between our algorithms and the following baseline algorithms to evaluate the effectiveness and efficiency of our proposed KT and KTIM algorithms.

- ◆ Greedy: Greedy is based on Algorithm 1 in this issue and utilizes the CELF algorithm in [8] to speed up the process.
- ◆ IMIT: IMIT [2] is based on the greedy algorithm, which optimizes the marginal gain calculation of nodes for seed selection.
- ◆ TIM: TIM [3] is a semi-heuristic and semi-greedy algorithm, which mainly contains a temporal heuristic stage and a temporal greedy stage.
- ◆ CCA: CCA [17] is a heuristic algorithm based on core level features of networks.
- ◆ DegreeDiscount: DegreeDiscount [10] is based on the degree of nodes. The basic idea is that when a node  $v$ 's neighbor node is selected as a seed node, its degree is discounted when calculating its degree for seed selection.

**Experimental environment.** The experimental platform is 64-bit Windows 10, the CPU is Intel Core i5-8250U @ 1.60 GHz quad-core, the memory is 8 GB, the hard disk is 128 GB, and the programming environment is Pycharm. All of the compared algorithms are implemented in python.

**Parameter settings.** We use the ICT model mentioned in section 3.3 to simulate the influence propagation. The seed set size is 10, 20, 30, 40, and 50, respectively. In the KT algorithm and KTIM algorithm, the size of the candidate seed set is set to 200. To pursue the accuracy of seed selection, the number of Monte Carlo simulations of the Greedy algorithm is set to 1000. The number of Monte Carlo simulations of the IMIT and TIM algorithms is set to 100. The radius of the influence parameter of the CCA algorithm is set to 1.

For the KTIM algorithm and the KT algorithm, because the network is layered by the improved K-shell method, the layered result at this stage is fixed. Therefore, this stage can be processed offline, and the running time of these two algorithms does not include the layered time in the experiments. For the IMIT and TIM algorithms, since 100 times Monte Carlo simulations in calculating the marginal gain are time-consuming, the data offline processing is also done. We do not consider the running time of Monte Carlo simulations when counting the running time for these two algorithms. Otherwise, the running time of IMIT and TIM would be much larger than the time without offline pre-processing.

## 5.2 The Efficiency of Different Algorithms

In this section, the running time of each algorithm is calculated for selecting 50 seed nodes in datasets of different sizes. Since the Greedy algorithm is time-consuming (The experiments on datasets 3 and 4 are unfinished for more than sixty hours) and its running time is much longer than that of the other algorithms, Greedy is not performed on datasets 3 and 4. The final results are shown in Table 2.

From Table 2, we have the following observations: First, KT, KTIM, and DegreeDiscount are more efficient than the other algorithms. Among them, KT performs best because it selects the node with the highest comprehensive degree as the seed node from each shell layer from core to edge. KTIM runs a little longer than KT since it first adds nodes close to the network core to the candidate seed set, and then sequentially selects the nodes with the highest comprehensive degree from the candidate seed set as seed nodes. Second, as the network scale increases, the running time of KT, KTIM, and DegreeDiscount increases very little because none of these algorithms are concerned with network



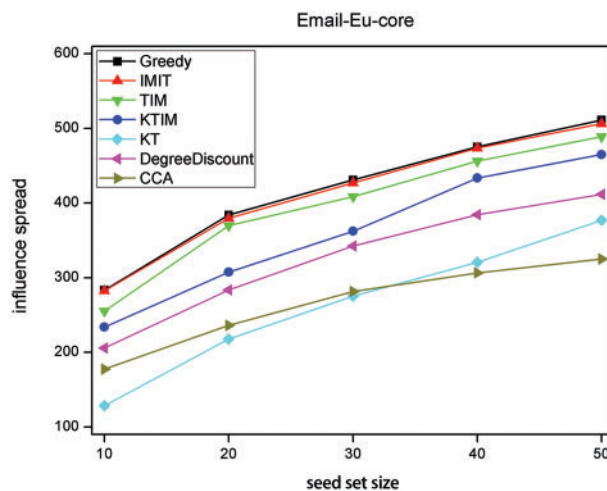
size in terms of seed selection. KT has the shortest running time followed by DegreeDiscount and KTIM, CCA and TIM are next, IMIT has a longer running time, and Greedy has the longest running time. Third, with the increase of the network size, the running time of the Greedy algorithm increases exponentially, which is actually unsuitable in temporal networks.

**Table 2:** Running time of the algorithms

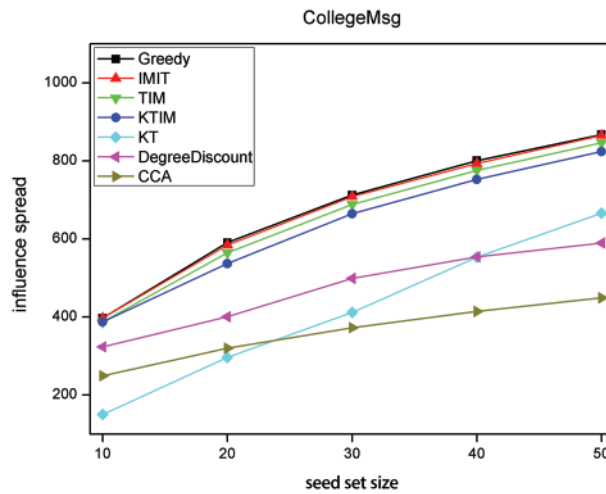
Algorithm	Email-Eu-core/s	CollegeMsg/s	Math overflow/s	Ask ubuntu/s
Greedy	72720	382222	/	/
IMIT	1.23	1.59	9.82	31.64
TIM	0.49	0.50	1.96	1.69
DegreeDiscount	0.03	0.24	0.26	0.14
CCA	0.64	2.19	3.17	1.28
KT	0.01	0.01	0.09	0.04
KTIM	0.11	0.071	0.36	0.24

### 5.3 The Effectiveness of Different Algorithms

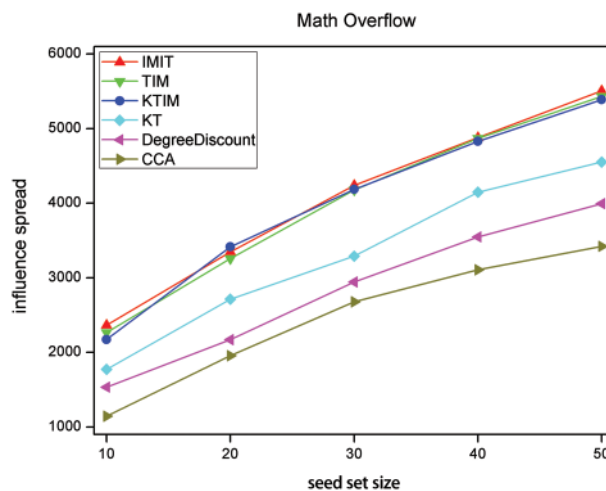
In this section, we test the effectiveness of the related algorithms. In each algorithm, when the seeds are selected, we run 1000 Monte Carlo simulations and take their average as a result. The final results are shown in Figs. 6 to 9. We should note that the larger the influence spread of the seed set, the better the algorithm's effectiveness. Since datasets 3 and 4 are too large and the running time of the Greedy algorithm is too high (unfinished more than sixty hours), the Greedy algorithm is not performed on these datasets. Fig. 6 shows the influence spread of seeds on the Email-Eu-core dataset. We can see that the influence spread of Greedy is larger than that of the other algorithms, followed by IMIT, TIM, and KTIM. TIM is slightly inferior to IMIT. Additionally, the gap between KTIM and TIM decreases as seed set size increases, and the performance of KT, DegreeDiscount, and CCA is lower than the other algorithms.



**Figure 6:** The results on the Email-Eu-core dataset



**Figure 7:** The results on the CollegeMsg dataset

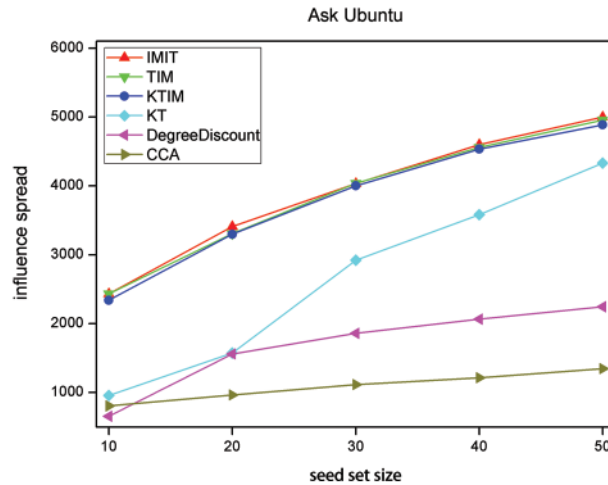


**Figure 8:** The results on the math overflow dataset

Fig. 7 shows the results on the CollegeMsg dataset. Greedy also has the best performance because it has performance guarantees. However, as seen in Table 2, the running time of Greedy is more than 100 h, which is useless in reality. The proposed KTIM algorithm is slightly inferior to IMIT and TIM in terms of effectiveness. While the running time of KTIM is much smaller than that of IMIT and TIM, e.g., the running time of KTIM is one and two orders of magnitude faster than that of TIM and IMIT, respectively. We also see that the performance of the KT algorithm improves rapidly as the seed set size increases, surpassing the DegreeDiscount algorithm and the CCA algorithm when  $k > 40$ . This is because the KT algorithm considers the temporal characteristic of nodes, starts with the global structure and local structure of the network, and combines the static and dynamic attributes of nodes so as to select the key nodes accurately.

From Figs. 8 and 9, we see that the influence range of IMIT, TIM, and KTIM is almost the same on the two datasets. However, as can be seen from Table 2, the running time of our proposed KTIM algorithm is much shorter than that of TIM and IMIT. Among the other algorithms, the KT algorithm

has a higher influence range in comparison to DegreeDiscount and CCA algorithms, which have a lower influence range as the seed set size increases. We can also see that the influence range of the KT algorithm in Fig. 9 is quickly expended for the same reason mentioned above.



**Figure 9:** The results on the ask ubuntu dataset

In short, according to Figs. 6 to 9, the Greedy algorithm performs best in terms of effectiveness. The performance of the KTIM, TIM, and IMIT algorithms is similar and close to that of the Greedy, followed by KT, and the DegreeDiscount and CCA algorithms are the worst.

#### 5.4 Analysis of Results

Throughout the analysis of the experimental results, Greedy has the largest influence but the longest running time, so this algorithm is not practical. Although DegreeDiscount and CCA have better performance in traditional influence maximization algorithms, they are not applicable to the IMT problem because they ignore the temporal characteristic of the network. In contrast, the KT and KTIM algorithms proposed in this paper, as well as the rest of the baseline algorithms, consider the network timing and are more suitable for the IMT problem. However, considering the efficiency and effectiveness of the algorithms, the KTIM algorithm achieves better results.

In terms of efficiency, as shown in Table 2, the KTIM algorithm is at least an order of magnitude faster than the IMIT algorithm and four times faster than the TIM algorithm. The reason is as follows, the time complexity of the KT algorithm is  $O(T + kn)$  and the KTIM algorithm is  $O(T + n + kN)$ . From reference [2,3], the time complexity of the IMIT algorithm is  $O(\varphi(v) |V| k + |V| \log_2 |V| k)$  and the TIM algorithm is  $O(n + kRk \log_2 |V| k)$ . Since  $N$  is the number of the candidate seeds, which is much smaller than the number of nodes  $|V|$ , so the KTIM algorithm has a shorter running time compared to the IMIT and TIM algorithms. More precisely, although IMIT optimizes the calculation of marginal gain, it still requires Monte Carlo simulations to calculate the influence, which takes a long time. Compared with IMIT, TIM reduces the running time by reducing all nodes in the network to candidate seed nodes, but it still requires a longer running time than KT and KTIM.

In terms of effectiveness, the KT algorithm proposed in this paper is slightly inferior to the rest of the temporal social network algorithms. However, the influence spread of the optimized KTIM algorithm is close to that of the IMIT algorithm and the TIM algorithm, which indirectly proves that the optimized seed selection strategy we propose is effective. Taking the Ask Ubuntu network in Fig. 9

as an example, when  $k = 50$ , the influence ranges of IMIT, TIM, and KTIM are 5100, 5000, and 4953, respectively, i.e., the influence spread of the KTIM algorithm is about 2.88% less than that of the IMIT algorithm and about 0.94% less than that of the TIM algorithm.

In conclusion, it is clear from the above analysis that the KTIM algorithm can trade a balance between effectiveness and efficiency. Therefore, considering the running time and influence range, KTIM has the best results and can efficiently solve the IMT problem.

## 6 Conclusion

In this work, we study the problem of maximizing the influence of temporal social networks. Although the greedy algorithm can solve this problem, it is time-consuming. To solve this problem, we consider both the global and local structures of temporal social networks. In the global scope, we propose an improved K-shell method to stratify the network according to the temporal characteristic of nodes and calculate the kernel value  $K_s$  of nodes to represent their static properties. In the local scope, we propose a calculation method of the comprehensive degree to weigh the influence of nodes in each core layer, and use the comprehensive degree to represent the dynamic properties of these nodes. On this basis, we propose the KT algorithm. It selects seed nodes with the highest comprehensive degree at each core layer from the central core to the edge core. Furthermore, we propose a more efficient KTIM algorithm by optimizing the seed selection strategy. It adds the nodes near the central core to the candidate seed set and then selects seed nodes with the highest comprehensive degree in the candidate set. Experimental results show that the KTIM algorithm runs at least an order of magnitude faster than the IMIT algorithm, has a similar influence range as it, and is suitable for temporal social networks. In the future, we will study the IMT problem in combination with more attributes, such as dynamic attributes and location attributes.

**Funding Statement:** This work is supported by the Youth Science and Technology Innovation Personnel Training Project of Heilongjiang (No. UNPYSCT-2020072), the Fundamental Research Funds for the Universities of Heilongjiang (Nos. 145109217, 135509234), the Education Science Fourteenth Five-Year Plan 2021 Project of Heilongjiang (No. GJB1421344) and the Innovative Research Projects for Postgraduates of Qiqihar University (No. YJSCX2022048).

**Conflicts of Interest:** The authors declare they have no conflicts of interest to report regarding the present study.

## References

- [1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. of the 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 57–66, 2001.
- [2] A. B. Wu, Y. Yuan, B. Y. Qiao, Y. S. Wang, Y. L. Ma *et al.*, "Research on algorithms for maximizing influence of large-scale time series diagrams," *Chinese Journal of Computers*, vol. 42, no. 12, pp. 2647–2664, 2019.
- [3] J. Chen and Z. Y. Qi, "Research on social network influence maximization algorithm based on time sequential relationship," *Journal on Communications*, vol. 41, no. 10, pp. 211–221, 2020.
- [4] L. Zhang and K. Li, "Influence maximization based on snapshot prediction in dynamic online social networks," *Mathematics*, vol. 10, no. 8, pp. 1341–1361, 2022.
- [5] J. Chandran and V. M. Viswanatham, "Dynamic node influence tracking based influence maximization on dynamic social networks," *Microprocessors and Microsystems*, vol. 95, no. 1, pp. 1–11, 2022.

- [6] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik *et al.*, “Identification of influential spreaders in complex networks,” *Nature Physics*, vol. 6, no. 11, pp. 88–93, 2010.
- [7] D. Kempe, J. Kleinberg and É. Tardos, “Maximizing the spread of influence through a social network,” in *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 137–146, 2003.
- [8] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen *et al.*, “Cost-effective outbreak detection in networks,” in *Proc. of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 420–429, 2007.
- [9] A. Goyal, W. Lu and L. Lakshmanan, “CELF++: Optimizing the greedy algorithm for influence maximization in social networks,” in *Proc. of the 20th Int. Conf. Companion on World Wide Web*, New York, NY, USA, pp. 47–48, 2011.
- [10] W. Chen, Y. Wang and S. Yang, “Efficient influence maximization in social networks,” in *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 199–208, 2009.
- [11] Y. Tang, X. Xiao and Y. Shi, “Influence maximization: Near-optimal time complexity meets practical efficiency,” in *Proc. of the 2014 ACM SIGMOD Int. Conf. on Management of Data*, New York, NY, USA, pp. 75–86, 2014.
- [12] Y. Tang, Y. Shi and X. Xiao, “Influence maximization in near-linear time: A martingale approach,” in *Proc. of the 2015 ACM SIGMOD Int. Conf. on Management of Data*, New York, NY, USA, pp. 1539–1554, 2015.
- [13] C. Borgs, M. Brautbar, J. Chayes and B. Lucier, “Maximizing social influence in nearly optimal time,” in *Proc. of the 25th Annual ACM-SIAM Symp. on Discrete Algorithms*, Portland, Oregon, USA, pp. 946–957, 2014.
- [14] W. Chen, C. Wang and Y. Wang, “Scalable influence maximization for prevalent viral marketing in large-scale social networks,” in *Proc. of the 16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 1029–1038, 2010.
- [15] K. Jung, W. Heo and W. Chen, “IRIE: Scalable and robust influence maximization in social networks,” in *Proc. of the 2012 IEEE 12th Int. Conf. on Data Mining*, Brussels, Belgium, pp. 918–923, 2012.
- [16] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. New York, NY, USA: Cambridge university press, pp. 169–219, 1994.
- [17] J. X. Cao, D. Dong, S. Xu, X. Zheng, B. Liu *et al.*, “A k-core based social network influence maximization algorithm,” *Chinese Journal of Computers*, vol. 38, no. 2, pp. 238–248, 2015.
- [18] S. Wang and J. Liu, “A memetic algorithm for solving the robust influence maximization problem towards network structural perturbances,” *Chinese Journal of Computers*, vol. 44, no. 6, pp. 1153–1167, 2021.
- [19] D. Wu, N. Q. He and X. H. Zhang, “Time-constraint influence maximization solution,” *Journal of Chinese Computer Systems*, vol. 43, no. 9, pp. 1–8, 2022.
- [20] C. Wang, Q. Shi, W. Xian, Y. Feng and C. Chen, “Efficient diversified influence maximization with adaptive policies,” *Knowledge-Based Systems*, vol. 213, no. 15, pp. 1–16, 2021.
- [21] J. Li, T. Sellis, J. S. Culpepper, Z. He, C. Liu *et al.*, “Geo-social influence spanning maximization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1653–1666, 2017.
- [22] L. Dash, B. K. Pattanayak, S. K. Mishra, K. S. Sahoo, N. Z. Jhanjhi *et al.*, “A data aggregation approach exploiting spatial and temporal correlation among sensor data in wireless sensor networks,” *Electronics*, vol. 11, no. 7, pp. 989–1006, 2022.
- [23] B. Vijayalakshmi, K. Ramar, N. Z. Jhanjhi, S. Verma, M. Kaliappan *et al.*, “An attention-based deep learning model for traffic flow prediction using spatiotemporal features towards sustainable smart city,” *International Journal of Communication Systems*, vol. 34, no. 3, pp. e4609, 2021.
- [24] A. Paranjape, A. R. Benson and J. Leskovec, “Motifs in temporal networks,” in *Proc. of the 10th ACM Int. Conf. on Web Search and Data Mining*, New York, NY, USA, pp. 601–610, 2017.

- [25] P. Panzarasa, T. Opsahl and K. M. Carley, "Patterns and dynamics of users' behavior and interaction: Network analysis of an online community," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 5, pp. 911–932, 2009.
- [26] T. Yasseri, R. Sumi and J. Kertész, "Circadian patterns of Wikipedia editorial activity: A demographic analysis," *PLoS One*, vol. 7, no. 1, pp. 1–8, 2012.