Information Sciences 602 (2022) 236–258

Contents lists available at ScienceDirect

Information Sciences

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

Efficient algorithms for finding diversified top-k structural hole spanners in social networks

Mengshi Li [a,1], Jian Peng [a,1], Shenggen Ju [a], Quanhui Liu [a], Hongyou Li [a], Weifa Liang [b], Jeffrey Xu Yu [c], Wenzheng Xu [a,*]

[a] College of Computer Science, Sichuan University, Chengdu 610065, PR China
[b] Department of Computer Science, City University of Hong Kong, 83 Tat Chee Ave., Kowloon, Hong Kong
[c] The Chinese University of Hong Kong, Hong Kong, PR China

ARTICLE INFO

Article history:
Received 20 August 2021
Received in revised form 20 January 2022
Accepted 24 April 2022
Available online 28 April 2022

Keywords:
Diversified hole spanners
Block information diffusion
Social network
Approximation algorithm

ABSTRACT

A structural hole spanner in a social network is a user who bridges multiple communities, and he can benefit from acting the bridging role, such as arbitrating information across different communities or getting earlier access to valuable and diverse information. Existing studies of finding hole spanners either identified redundant hole spanners (i.e., communities bridged by different hole spanners are redundant) or found nonredundant hole spanners only by network structure. Unlike the existing studies, we not only study a problem of finding top-k hole spanners that connect nonredundant communities in the social network, but also consider the tie strengths between different pairs of users and the different information sharing rates of different users, so that after removing the found users, the number of blocked information diffusion is maximized. In addition, we devise a novel $(1-\frac{1}{e})$-approximation algorithm for the problem, where e is the base of the natural logarithm. We further propose a fast randomized algorithm with a smaller time complexity. Our experiment results demonstrate that, after removing the nodes found by the proposed two algorithms, the numbers of blocked information diffusion can be up to 80% larger than those by existing algorithms.

1. Introduction

Social networks (e.g., Weibo, WeChat, Twitter, Facebook, etc.) have experienced exponential growth in the last decades and become the most popular and effective tools for sharing information. Users in such networks can quickly and conveniently exchange information with their friends. The users who share highly similar attributes, such as similar education backgrounds, interests, ideas, etc., form the so-called community structure [9,15,19,24,30,42], where the users are tightly linked with each other within their community, but only loosely linked with users in other communities. That is, a user contacts much more frequently with the users in his/her community than the users outside his/her community. As a result, the information shared by the users within the same community may be redundant or homogeneous, whereas the information received from the users in different communities are probably nonredundant.

[*] Corresponding author.
E-mail addresses: aviva_s@163.com (M. Li), jianpeng@scu.edu.cn (J. Peng), jsg@scu.edu.cn (S. Ju), quanhuiliu@scu.edu.cn (Q. Liu), lihongyou@scu.edu.cn (H. Li), weifa.liang@cityu.edu.hk (W. Liang), yu@se.cuhk.edu.hk (J.X. Yu), wenzheng.xu3@gmail.com (W. Xu).
[1] The first two authors Mengshi Li and Jian Peng contributed equally to this work, i.e., co-primary authors.

https://doi.org/10.1016/j.ins.2022.04.046
0020-0255/© 2022 Elsevier Inc. All rights reserved.

A *structural hole spanner* in a social network is a user who bridges multiple communities, and he can benefit from acting the bridging role [5–7,10,28]. For example, because such a user connects with multiple nonredundant communities, he/she can get earlier access to valuable and diverse information from different communities than another user who only belongs to his/her community, or arbitrate information across different communities.

The identification of top-k hole spanners has many valuable applications in social networks. For example, it has been shown that misinformation, such as rumors, spread six times faster than the truth in social networks [37]. Then, without any intervention, the misinformation may spread to many users in a short time, and thus incur negative effects, such as hurting a person's pride, and bringing anxiety and depression to some persons [1,4,26,38]. Fortunately, we can effectively prevent the wide diffusions of misinformation by finding only a few structural hole users and quarantining the misinformation received by them. Another application of structural hole spanners arises in academic collaboration networks [6,18,39], where the hole spanners are the researchers who have research interests in different fields. They are able to generate innovative ideas by combining knowledge from the different fields, or solve an important but challenging problem faced in one field by applying innovative technologies and ideas from other fields. Other applications of structural hole spanners include community kernel detection, link prediction, specific user group classification, etc [25,43].

1.1. Motivations

Several studies have been proposed for identifying structural hole spanners. Most of the studies identified top-k *redundant* hole spanners [5,11,16,17,20,25,36,39]. That is, they first measured each node with a score for acting as a hole spanner. They then found the k nodes with the maximum or minimum scores to be the top-k hole spanners. It can be observed that although each of these found hole spanners may have a high score for acting the bridge role, it is possible that the communities spanned by them are redundant. Then, the benefit obtained by them may be much less than the sum of benefits obtained by each of them. This indicates that the benefit obtained by them is overestimated.

For instance, Fig. 1 shows a social network with four structural hole spanners v_1, v_2, v_3, and v_4, and six communities $C_1, C_2, \ldots, C_6$, where thick and thin lines indicate strong and weak connections between users, respectively. To identify the top-2 hole spanners for preventing the widespread of misinformation, e.g., rumors, in the network, the existing studies in [5,11,16,17,20,25,36,39] may detect users v_1 and v_2, since each of them connects with three communities, while user v_3 only spans two communities and user v_4 has weak connections with his connected three communities. However, it can be seen that communities C_1 and C_3 are spanned by both users v_1 and v_2. Thus, the benefit obtained by filtering misinformation passing through users v_1 and v_2 may be limited.

We notice that there are only a very few studies that identified *nonredundant* hole spanners [25,32,40] by counting the number of communities bridged by the identified hole spanners. Then, they may detect v_1 and v_4 as the top-2 hole spanners, since they bridge the six nonredundant communities. However, the tie strengths between user v_4 and his bridged communities C_4, C_5 and C_6 are weak, which means that only limited pieces of misinformation are diffused among the communities C_4, C_5 and C_6 through user v_4. Then, only limited pieces of misinformation diffusion will be blocked by quarantining user v_4.

In spite of the pioneering studies in [25,32,40] to identify diversified (i.e., nonredundant) structural hole spanners, they only exploited the network structure of hole spanners, and did not consider two other important things. One is that they ignored the tie strengths between different pairs of users. Following the recent study of Burt [8], the profit of acting a bridging role obtained by a hole spanner is proportional to the tie strengths with his connected communities, where the tie strength between the hole spanner and one of the communities is measured by the number of friends in the community and the interaction frequencies with the friends. That is, the stronger ties with his connected communities a hole spanner has, the higher profit the hole spanner can obtains. The rationale behind is that, if a user has stronger ties with his connected communities, he is more likely to obtain valuable information from members in the communities. In addition, the information sent by him is more easily to be trusted by the community members [12,34]. For example, although users v_1 and v_3 span only five communities, less than the six communities spanned by users v_1 and v_4, both users v_1 and v_3 establish strong ties with their bridged communities, which means that large numbers of misinformation are diffused through v_1 and v_3. Therefore, more misinformation diffusion will be blocked by quarantining users v_1 and v_3, compared with users v_1 and v_4.

The other important thing is that existing studies did not consider that only a small portion of users are very active in sharing their information in a social network, while most of the users are inactive. The famous "90-9-1" rule says that 90% users only observe and/or read information but not create, 9% users create a little information, while 1% users actively contribute new information [2,27,35]. Therefore, different users have significantly different information sharing rates, whereas most existing studies assumed that the sharing rates of different users are identical [5,11,16,17,20,25,32,36,40].

Unlike the existing studies that either identified only redundant hole spanners [5,11,16,17,20,25,36,39] or found nonredundant hole spanners only by network structure [25,32,40], in this paper we not only identify top-k hole spanners that connect nonredundant communities, but also consider the tie strengths between different pairs of users and the different information sharing rates of different users, such that, after removing these found hole spanners, the number of blocked information diffusion in social networks is maximized. Following our experiments later, users v_1 and v_3 in Fig. 1 will be identified as the top-2 hole spanners.

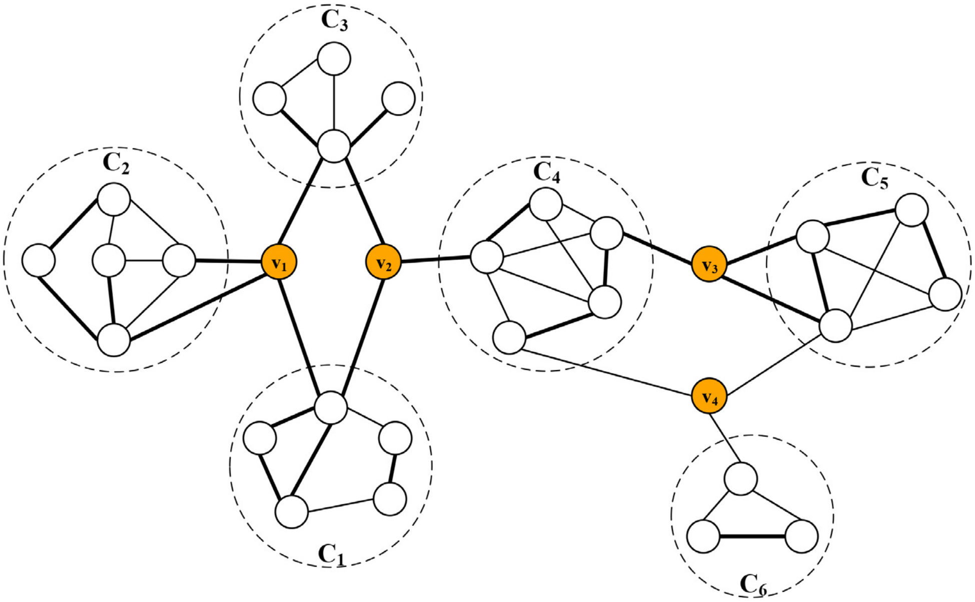


Fig. 1. An illustration of top-2 diversified hole spanners, where there are four hole spanners v_1, v_2, v_3, and v_4 and six communities $C_1, C_2, \ldots, C_6$ in a social network, a thick between two users indicate a strong contact strength between them while a thin line implies a weak contact strength.

1.2. Contributions

We focus on developing efficient algorithms to identify diversified hole spanners in social networks. We summarize the contributions of this work.

- We consider the identification of top-k diversified hole spanners which connect nonredundant communities in social networks. Also, we consider the tie strengths between different pairs of users and the different information sharing rates of different users.
- We study a problem of finding top-k diversified hole spanners, which is to find k users a social network and remove them from the network, such that the number of blocked information diffusion is maximized.
- We devise a $(1-\frac{1}{e})$-approximation algorithm with a time complexity of $O\left(knm + kn^2 \log n\right)$ for the problem, where e is the base of the natural logarithm, n is the number of users, and m is the number of connections in the social network.
- We propose a fast randomized algorithm for the problem, based on the new sampling technique, which has a much smaller time complexity than the approximation algorithm.
- We evaluate the algorithm performance with real-world network datasets. Our experiment results demonstrate that, after removing the hole spanners found by the two proposed algorithms, the numbers of blocked information diffusion can be up to 80% larger than those by existing algorithms. Moreover, the number of blocked information diffusion by the randomized algorithm is only slightly less than that by the approximation algorithm, while the randomized algorithm is from 1.75 to 100 times faster than the approximation algorithm.

The paper organization of the rest paper is as follows. In Section 2, we review related work. In Section 3, we introduce system models and define the problem. In Section 4, we devise an approximation algorithm for the problem, and analyze its performance. In Section 5, we propose a fast randomized algorithm for the problem. In Section 6, we empirically evaluate the algorithm performance. Finally, in Section 7, we conclude this work.

2. Related work

The identification of a few but important structural hole spanners to block the diffusion of false news and rumors has wide applications, with the exponential growth of social networks over the past decades. The task however is very challenging due to the large-scale of such networks. Most existing algorithms identified top-k redundant hole spanners. That is, they first measured each node with a score for acting as a hole spanner. They then selected the k nodes with the maximum or minimum scores as the top-k hole spanners. Although hole spanners found by the algorithms have high scores for blocking information diffusion, it is possible that the communities spanned by them are redundant, and the number of blocked information diffusion by them may be much less than the sum of blocked information diffusion by them. The effect on blocking information diffusion by them thus is overestimated.

The studies on the identification of *redundant* hole spanners are briefly introduced as follows. Burt [5] proposed two metrics to rank players in an entrepreneurial networks by measuring their capabilities of acquiring potential resources from their spanned communities. Then, the top-k hole spanners are the players with the largest scores measured by these metrics. Specifically, Burt proposed two metrics of effective size and network constraint. On one hand, the effective size measures the number of nonredundant contacts in the ego network of a player. On the other hand, the network constraint measures the amount of time that a player spends for redundant contacts, where two contacts of the player are redundant if they are friends of each other. Then, the higher constraint value the player has, the less opportunity that the player acts as a hole spanner. Goyal et al. [17] assumed that the benefit of a node for acting as an information diffusion intermediator is proportional to the number of shortest paths on which the node lies. Since it usually takes a long time to search shortest paths between all pairs in a large graph, Tang et al. [36] devised a simple algorithm to find hole spanners by counting only the shortest paths with two steps passing through each node, and selected the top 1% nodes. Lou et al. [25] modeled hole spanners by utilizing the theory of two-step information flow, which says that information usually first flows to an opinion leader, then flows from the opinion leader to many other persons. They assumed that communities are given, and a user is assigned a higher score for acting as a hole spanner if the user connects more opinions leaders in different communities. He et al. [20] assumed that each user is contained in only one community and the number of communities is given. Then, they proposed a harmonic modularity scheme to find communities and structural hole spanners simultaneously. Chang et al. [11] modeled structural hole spanners by structural diversities, and a node's structural diversity is defined as the number of connected components in its ego network. They then selected the k nodes with highest structural diversities as hole spanners. Xu et al. [39] observed that important hole spanners are the users who not only connect with diverse communities, but also build close relationships with these connected communities. They measured the importance of each node as a hole spanner by the number of blocked information diffusion if removing the node from the network, then the k nodes with the maximum values are the top-k hole spanners. They proposed novel approximation algorithms. Based on a previous work [41] on community detection using community forest model, Zhang et al. [44] observed a phenomenon of diminishing marginal utility in the process of community reconstruction. Motivated by this observation, they proposed a method to identify hole spanners in social networks. They identified two types of hole spanners: the first type of hole spanners are the users that violate the law of diminishing marginal utility and thus change the trend of community expansion, while the other type of hole spanners are the users that belong to multiple communities. They then ranked the influence of the two types of hole spanners, and selected the most influential nodes as the top-k hole spanners. Furthermore, some study based on machine learning has been introduced recently. Specifically, Gong et al. [16] first identified hole spanners by utilizing the algorithm in [25], then learned the characteristics of the spanners with machine learning. Once a new user joins the social network, they can determine whether the user is a new hole spanner by the characteristics of the user, without running the algorithm in [25] again. However, they did not rank hole spanners.

We noticed that there are a very few existing studies that identified *diversified* hole spanners, which means that the communities spanned by them are highly nonredundant. For example, Lou et al. [25] characterized hole spanners as the users who bridge the holes control the information diffusion between different communities. They assumed that ground-truth communities are given, and considered the top-k hole spanners as the k nodes such that, after the removing the k nodes, the decrease of the minimal cut in a network is maximized, where the minimal cut of communities in a network is the minimum number of edges so that different communities separate from each other after the removals of the edges. However, the performance of the found hole spanners depends on the quality of given communities. Xu et al. [40] modeled the top-k hole spanners by finding k nodes such that the increase of the communication cost of a network is maximized by their removals, where the communication cost is the sum of distances of shortest paths between all pairs of nodes. They then devised two efficient algorithms for detecting hole spanners. However, they identified top-k diversified hole spanners by only exploiting network structure.

Although the algorithms in [25,40] can be used to find diversified hole spanners, they did not consider two important things. Following the recent study of Burt [8], the profit of acting a bridging role obtained by a hole spanner is proportional to the tie strengths with his connected communities, where the tie strength between the hole spanner and one of the communities is measured by the number of friends in the community and the interaction frequencies with the friends. That is, the stronger ties with his connected communities a hole spanner has, the higher profit the hole spanner can obtains. The rationale behind is that, if a user has stronger ties with his connected communities, he is more likely to obtain valuable information from members in the communities. In addition, the information sent by him is more easily to be trusted by the community members [12,34].

The other important thing is that existing studies did not consider that only a small portion of users are very active in sharing their information in a social network, while most of the users are inactive. The famous "90-9-1" rule says that 90% users only observe and/or read information but not create, 9% users create a little information, while 1% users actively contribute new information [2,27,35]. Therefore, different users have significantly different information sharing rates, whereas most existing studies assumed that the sharing rates of different users are identical [5,11,16,17,20,25,32,36,40].

Different from the aforementioned existing studies, we identified the top-k hole spanners by considering not only the hole spanners that connect diversified communities, but also the tie strengths between different pairs of users and the different information sharing rates of different users, so that the number of information diffusion blocked by the found hole spanners is maximized.

3. Preliminary

We first present the network model, and the information diffusion model. We then define the problem.

3.1. Network model

We consider a social network $G = (V, E)$, which is a directed graph, where V is the set of users in the network, and E is the set of edges representing connections between different users. Let n be the number of users, and m be the number of edges in network G.

Consider each directed edge (v_i, v_j) in network G. Let w_{ij} be the information diffusion probability from user v_i to v_j, and its value indicates the extent of user v_j being influenced by user v_i. We can estimate the value of w_{ij} from historical information in the following way. If user v_i has already sent c_{ij} numbers of information to v_j, whereas v_j only forwarded $c\prime_{ij}$ numbers of information to his friends with $c\prime_{ij} \leqslant c_{ij}$. Then, the value of w_{ij} is estimated as $w_{ij} = \frac{c\prime_{ij}}{c_{ij}}$. Clearly, $0 \leqslant w_{ij} \leqslant 1$. Notice that the information diffusion probability w_{ij} from users v_i to v_j and the probability w_{ji} from users v_j to v_i usually are different.

Denote by f_i the average information sharing rate of each user $v_i \in V$, where user v_i shares his information with his friends, e.g., photos, videos, ideas, interests, where $f_i > 0$. We assume that $f_i \leqslant 1$ for each user v_i. Otherwise ($f_i > 1$ for some users), the average information sharing rate f_i can be normalized by dividing the maximum rate f_{max} in the network, where $f_{max} = \max_{v_j \in V}\{f_j\}$.

3.2. Information diffusion model

Given any two different users s and t in network G, and a simple path $P_j =< s, v_1, v_2, \ldots, v_{n_j}, t >$ from user s to user t, the success probability p_j^{st} that a piece of information sent by user s is received by user t via path P_j is

$$p_j^{st} = \prod_{i=0}^{n_j} w(v_i, v_{i+1}), \tag{1}$$

where n_j is the number of intermediate nodes in path $P_j, w(v_i, v_{i+1})$ is the information diffusion probability from user v_i to user $v_{i+1}, v_0 = s$, and $v_{n_j+1} = t$. Denote by $\mathscr{P}_{st}$ the set of simple paths from s and t in G. Also, let P_{st}^{max} be the path in $\mathscr{P}_{st}$ with the maximum diffusion probability, i.e.,

$$P_{st}^{max} = \arg\max_{P_j \in \mathscr{P}_{st}}\left\{p_j^{st}\right\}. \tag{2}$$

For example, Fig. 2(a) demonstrates four simple paths from v_1 to v_6 in network G, where $P_1 = \langle v_1, v_2, v_6\rangle$, $P_2 = \langle v_1, v_2, v_3, v_6\rangle, P_3 = \langle v_1, v_3, v_6\rangle$ and $P_4 = \langle v_1, v_4, v_6\rangle$. It can be seen that $P_3 = P_{st}^{max} = \langle v_1, v_3, v_6\rangle = w_{1,3} \cdot w_{3,6} = 0.4 \cdot 0.6 = 0.24$.

We assume that a piece of information diffuses along the path P_{st}^{max} with maximum diffusion probability from s to t due to the following two reasons. The first reason is that a piece of misinformation, e.g., rumors, may spread to a large number of users in a social network in a short time [37]. The path with the maximum diffusion probability usually is shorter than the paths with smaller diffusion probabilities, and it thus takes shorter time for the information to propagate along the path. In order to quickly prevent the diffusion of the information to a large number of users, we find hole spanners on the path with the maximum diffusion probability.

The second reason is that many existing studies assumed that information spread along shortest paths and ignored the information diffusion probability between users. Freeman [13] proposed the metric of "betweenness centrality". This metric calculates the numbers of shortest paths passing through each node to evaluate its influence. The more information passing through a node, the higher betweenness centrality the node has, assuming that information diffuses along the shortest paths. Pinto et al. [31] considered a problem of finding the source node of an information diffusion from the knowledge of only a portion of intermediate users in the information diffusion paths, assuming that the information only diffuses along the shortest path from the source to intermediate users. Rodriguez et al. [33] measured the network centrality of online medias over time on different hot topics through computing the shortest path length between sites in the network. The site with a high centrality typically lies at the network "center", assuming that the information diffusion pathways of topics in the online media space are the shortest path. Kimura and Saito [22] recommended two information diffusion models to approximate the well-known Independent Cascade Model, so as to quickly measure influence of diffusions, which assumes that each node can only be activated when it is on the shortest paths from source nodes.

3.3. Find paths with the maximum information diffusion probabilities

Given a source node s, the paths with the maximum information diffusion probabilities from s to each node in set $V \setminus \{s\}$ can be found in the following way. First, a graph $G\prime = (V, E)$ with its information diffusion probability function $w\prime : E \mapsto \mathbb{R}^+$ is constructed from graph G, where the weight $w\prime_{ij}$ of each edge (v_i, v_j) in $G\prime$ is set as $w\prime_{ij} = \log_2 \frac{1}{w_{ij}}$, and w_{ij} is the information diffusion probability from v_i to v_j. For instance, Fig. 2(b) demonstrates such a graph $G\prime$ constructed from G in Fig. 2(a).

The shortest paths in $G\prime$ from s to the other nodes in set $V \setminus \{s\}$ then can be found by applying Dijkstra's algorithm. Denote by d_{s,v_i} the shortest distance in graph $G\prime$ from s to v_i. Also, it can be seen that the union of the shortest paths is a spanning tree T_s of $G\prime$ rooted at s. For example, Fig. 2(c) shows the shortest path tree $T_1\prime$ in $G\prime$ of Fig. 2(b) rooted at v_1, where the value next to each node v_i in $T_1\prime$ is the shortest distance from the source v_1 to v_i.

Finally, a tree T_s in the original graph G rooted at s is obtained, which has the same structure with $T_s\prime$, i.e., $T_s = T_s\prime$. The maximum diffusion probability p_{s,v_i} in T_s from s to v_i is $p_{s,v_i} = \frac{1}{2^{d_{s,v_i}}}$, where d_{s,v_i} is the shortest distance from s to v_i in graph $G\prime$. For instance, Fig. 2(d) demonstrates the *maximum likely diffusion tree* T_1 in G rooted at v_1, where the value next to each node v_i in T_1 is the maximum diffusion probability from v_1 to v_i.

Procedure 1 presents the process of finding the paths with the maximum diffusion probabilities from the source node s to nodes in $V \setminus \{s\}$.

Procedure 1: `Find paths with the maximum diffusion probabilities.`

```
Input: A social network $G = (V, E)$, an information diffusion probability function $w : E \mapsto [0, 1]$, and a source node $s$ in $V$
Output: The paths with the maximum diffusion probabilities from the source node $s$ to nodes in $V \setminus \{s\}$
Construct a graph $G\prime = (V, E)$ with its information diffusion probability function $w\prime : E \mapsto \mathbb{R}^+$ from $G$ such that
  $w\prime_{i,j} = \log_2 \frac{1}{w_{ij}}$;
for $i \leftarrow 1$ to $n$ do
    $d_{s,v_i} \leftarrow \infty$;/* shortest distance in $G\prime$ from $s$ to $v_i$ */
    $p_{s,v_i} \leftarrow 0$;/* the maximum diffusion probability in $G$ from $s$ to $v_i$ */
end for
Find the shortest paths from $s$ to each node $v_i \in V \setminus \{s\}$ by running Dijkstra's algorithm, and obtain the values of $d_{s,v_i}$;
Obtain a tree $T\prime_s$ in $G\prime$ rooted at node $s$ by the union of shortest paths from $s$;
Obtain a maximum likely diffusion tree $T_s$ in $G$ rooted at $s$ from $T\prime_s$ with $p_{s,v_i} = \frac{1}{2^{d_{s,v_i}}}$;
```

We claim that the path from s to t in tree T_s is the path with the maximum diffusion probability in G, which can be proved by the following lemma.

Lemma 1. *Consider a social network $G = (V, E)$ with its information diffusion probability function $w : E \mapsto [0, 1]$, let $G\prime = (V, E)$ be a graph constructed from network G, where the information diffusion probability function $w\prime : E \mapsto \mathbb{R}^+$ in $G\prime$ is $w\prime_{ij} = \log_2 \frac{1}{w_{ij}}$. Assume that a path P_{st} in $G\prime$ is the shortest path from s to t with $s, t \in V$. We claim that P_{st} is the path in G from s to t with the maximum diffusion probability.*

Proof. See A. □

Lemma 2. *Given a social network $G = (V, E)$, and an information diffusion probability function $w : E \mapsto [0, 1]$, there is an algorithm, i.e., Procedure 1, to find paths with the maximum diffusion probabilities from a given source node s in G with a time complexity of $O(m + n \log n)$, where $n = |V|$ and $m = |E|$.*

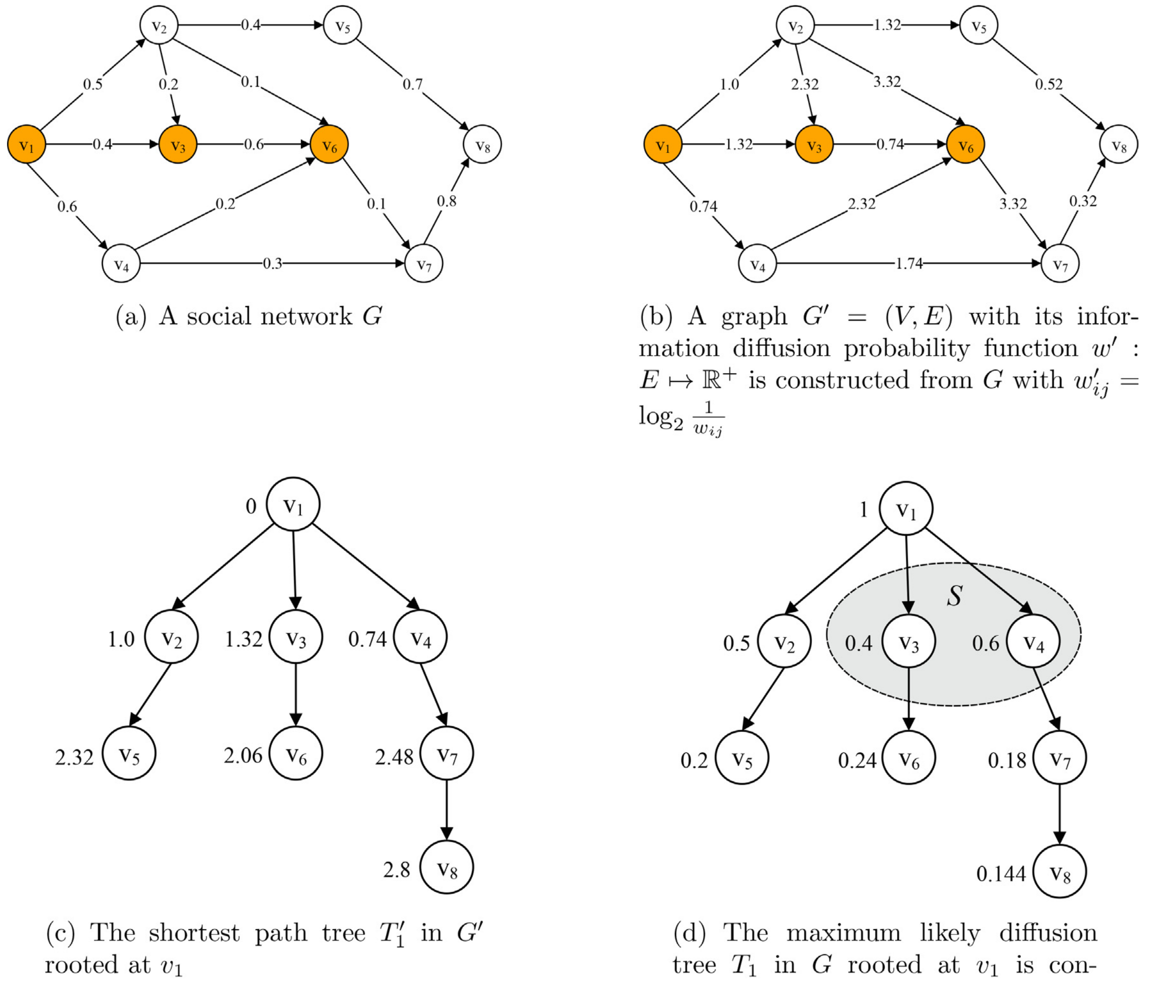
Proof. It can be seen that the time complexity of Procedure 1 is dominated by the execution of Dijkstra's algorithm, which can be implemented in a time complexity of $O(m + n \log n)$ [14]. □

3.4. Problem definition

Consider the maximum likely diffusion tree T_j rooted at any source node $s_j \in V$, e.g., T_1 rooted at v_1 in Fig. 2(d). We can see that once removing a node v_i in T_j, the descendants of v_i in T_j will not receive the information diffused from the source node s_j. Denote by $D_j(v_i)$ the set of descendants of node v_i in T_j, and $D_j(v_i)$ also is the set nodes in the subtree rooted at v_i in T_j. For instance, consider the subtree rooted at v_4 in Fig. 2(d), we have $D_1(v_4) = \{v_4, v_7, v_8\}$.

Given any potential set $S(\subseteq V)$ of structural hole spanners in G, denote by $D_j(S)$ the set of descendants of a node in S in T_j, i.e., $D_j(S) = \bigcup_{v_i \in S} D_j(v_i)$. For example, assume that $S = \{v_3, v_4\}$, then $D_j(S) = \{v_3, v_4, v_6, v_7, v_8\}$, see Fig. 2(d).

For each node $v_l \in T_j \setminus \{s_j\}$, it can be seen that the number of diffused information from s_j to v_l per unit time is $p_{jl} \cdot f_j$, where p_{jl} is the maximum information diffusion probability from s_j to v_l, and f_j is the information sharing rate of the source s_j. Then, after removing the nodes in set S in tree T_j, the number of blocked information diffusion is



(a) A social network G

(b) A graph $G' = (V, E)$ with its information diffusion probability function $w' : E \mapsto \mathbb{R}^+$ is constructed from G with $w'_{ij} = \log_2 \frac{1}{w_{ij}}$

(c) The shortest path tree T'_1 in G' rooted at v_1

(d) The maximum likely diffusion tree T_1 in G rooted at v_1 is constructed from T'_1

Fig. 2. An illustration of the information diffusion model.

$$H_j(S) = f_j \sum_{v_l \in D_j(S)} p_{jl}. \tag{3}$$

Denote by $T_1, T_2, \ldots, T_n$ the maximum likely diffusion trees rooted at $v_1, v_2, \ldots, v_n$ respectively. Then, after removing the nodes in set S, the number of blocked information diffusion in the social network G is

$$H(S) = \sum_{j=1}^{n} H_j(S) = \sum_{j=1}^{n} f_j \sum_{v_l \in D_j(S)} p_{jl}. \tag{4}$$

Given a social network $G = (V, E)$, the information diffusion probability function $w : E \mapsto [0, 1]$, and the information sharing rate function $f : V \mapsto \mathbb{R}^+$, *the top-k diversified hole spanners problem* in G is defined as to find a set S of k nodes in G with $k \leqslant |V|$, so that, after removing the nodes in S, the number of blocked information diffusion is maximized, i.e.,

$$\text{maximize}_{S \subseteq V, |S|=k} \{H(S)\}. \tag{5}$$

3.5. Notion of submodular functions

Let V be set of nodes and H be a function with $H : 2^V \mapsto \mathbb{R}^{\geqslant 0}$. $H(.)$ is a nondecreasing submodular function, if the function meets the following properties.

(i) $H(\varnothing) = 0$;

(ii) Monotonicity: $H(A) \leqslant H(B)$ for any two subsets A and B of V with $A \subseteq B$;

(iii) Submodularity: $H(A \cup \{v\}) - f(A) \geqslant H(B \cup \{v\}) - f(B)$ for any two subsets A and B of V with $A \subseteq B$, and any node v in $V \setminus B$.

4. Proposed approximation algorithm

We devise a $(1 - \frac{1}{e})$-approximation algorithm for the top-k diversified hole spanner problem and its time complexity is $O\left(knm + kn^2 \log n\right)$, where e is the base of the natural logarithm, $n = |V|$, and $m = |E|$.

The proposed approximation algorithm identifies the top-k hole spanners by adopting a greedy strategy. Let S be the set of hole spanners detected by the algorithm. Initially, $S = \varnothing$. Within the ith iteration $(1 \leqslant i \leqslant k)$, the algorithm identifies a node u_i in set $V \setminus S$ that blocks the maximum marginal number of information diffusion, i.e., $u_i = \arg\max_{v \in V \setminus S}\{H(S \cup \{v\}) - H(S)\}$. The procedure continues until the number of nodes in S reaches k.

In the following, we start by introducing the process of identifying the next hole spanner u_i. We then present the approximation algorithm. We finally analyze the algorithm performance.

4.1. Approximation algorithm

The key in the proposed algorithm is to identify the next hole spanner that blocks the maximum marginal number of information diffusion. Let $S = \{u_1, u_2, \cdots, u_{i-1}\}$ be the set of hole spanners identified within the first $(i-1)$ iterations. To identify the next spanner u_i, we show how to compute the marginal number of blocked information diffusion after removing each node v in set $V \setminus S$.

Given a network $G = (V, E)$ with its information diffusion probability function $w : E \mapsto [0, 1]$, for each source node $s_j \in V$, we first obtain the maximum likely diffusion tree T_j in G originated from s_j by invoking Procedure 1 in Section 3.3. For instance, Fig. 3(b) demonstrates a maximum likely diffusion tree T_1 from source node v_1, where T_1 is constructed from the network G in Fig. 3(a).

Recall that $D_j(S)$ represents the set of descendants of nodes in S in tree T_j. For example, let $S = \{v_8, v_9\}$, we have that $D_1(S) = \{v_8, v_{12}, v_{13}, v_{14}, v_9, v_{15}\}$ in Fig. 3(b).

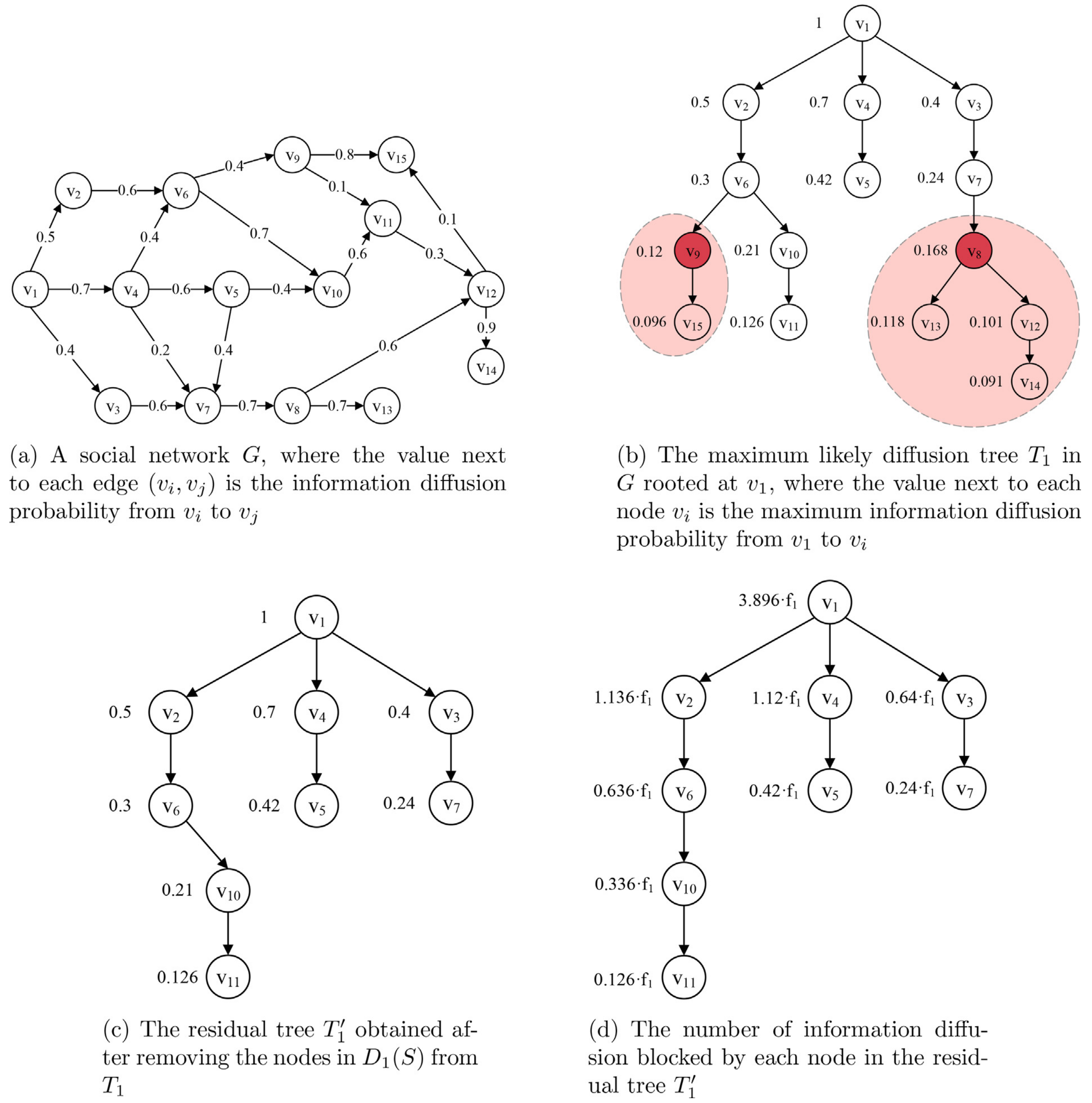
Algorithm 1: Identify the next diversified hole spanner.

```
Input: A social network $G' = (V, E)$, an information diffusion probability function $w' : E \mapsto \mathbb{R}^+$, and a set $S \in V$ of nodes
  with $|S| < k$
Output: Identify a node $u_i$ such that the marginal number of blocked information diffusion after removing the node is
  maximized
For each $v_l \in V$, let $H'(v_l) \leftarrow 0$;/* the marginal number of blocked information diffusion after removing each node $v_l$ */
for $j \leftarrow 1$ to $n$ do
    Find the maximum likely diffusion tree $T_j$ in $G$ rooted at source node $s_j$, by invoking Procedure 1 in Section 3.3;
    Obtain a tree $T'_j$ by removing the nodes in $D_j(S)$ from $T_j$;
    Calculate the marginal number $H'_j(v_l)$ of blocked information diffusion of each node $v_l \in T'_j$ by performing a DFS
      starting from the source node $s_j$;
    foreach node $v_l \in T'_j$ do
        $H'(v_l) \leftarrow H'(v_l) + H'_j(v_l)$;
    end for
end for
return a node $u_i$ so that $H'(u_i) = \max_{v_l \in V \setminus S}\{H'(v_l)\}$.
```

We then obtain a tree T'_j by the removals of the nodes in $D_j(S)$ from T_j, since the removals of nodes in S have already blocked the information diffusion from s_j to nodes in $D_j(S)$. For example, Fig. 3(c) shows the residual tree after removing the nodes in $D_1(S)$ from T_1.

Finally, we calculate the marginal number of blocked information diffusion after removing each node $v_l \in T'_j$. Specifically, since the removal of a node v_l in T'_j can block the information diffusion from the source node s_j to the descendants of node v_l in T'_j, the marginal number of blocked information diffusion after removing a node $v_l \in T'_j$ is the sum of information diffusion received by nodes in $D'_j(v_l)$, where $D'_j(v_l)$ represents the set of descendants of v_l in tree T'_j. Then, the marginal number $H'_j(v_l)$ of information diffusion blocked by node v_l in tree T'_j is

$$H'_j(v_l) = f_j \cdot \sum_{v_t \in D'_j(v_l)} p_{jt}, \tag{6}$$



(a) A social network G, where the value next to each edge (v_i, v_j) is the information diffusion probability from v_i to v_j

(b) The maximum likely diffusion tree T_1 in G rooted at v_1, where the value next to each node v_i is the maximum information diffusion probability from v_1 to v_i

(c) The residual tree T_1' obtained after removing the nodes in $D_1(S)$ from T_1

(d) The number of information diffusion blocked by each node in the residual tree T_1'

Fig. 3. An illustration of identifying the next hole spanner.

where f_j is the information sharing rate of the source s_j, and p_{jt} is the maximum information diffusion probability from s_j to v_t. For example, Fig. 3(d) shows that $D\prime_1(v_6) = \{v_6, v_{10}, v_{11}\}$. The marginal number of blocked information diffusion after removing node v_6 in tree $T\prime_1$ then is $H\prime_1(v_6) = f_1 \cdot (p_{1,6} + p_{1,10} + p_{1,11}) = f_1 \cdot (0.3 + 0.21 + 0.126) = 0.636 \cdot f_1$. In addition, the marginal number of blocked information diffusion after removing each node $v_l \in T\prime_j$ can be computed with a Depth-First Search (DFS), by starting from the source node s_j.

Notice that different nodes in network G share their information at different rates. Then, the marginal number of blocked information diffusion after removing v_l in network G is the sum of marginal number of blocked information diffusion after removing v_l in the n diffusion trees $T_1, T_2, \ldots, T_n$ originated from the source nodes $s_1, s_2, \ldots, s_n$ in G, respectively, i.e.,

$$H\prime(v_l) = \sum_{j=1}^{n} H\prime_j(v_l) = \sum_{j=1}^{n} f_j \cdot \sum_{v_t \in D\prime_j(v_l)} p_{jt}. \tag{7}$$

The next hole spanner u_i then is a node such that its removal blocks the maximum marginal number of information diffusion in G, i.e., $u_i = \arg\max_{v_l \in V \setminus S}\{H\prime(v_l)\}$. The algorithm for identifying the next hole spanner can be found in Algorithm 1.

The approximation algorithm is presented in Algorithm 2.

Algorithm 2: Approximation algorithm (DHS).

```
Input: A social network $G = (V, E)$, an information diffusion probability function $w : E \mapsto [0, 1]$, an information sharing
  rate function $f : V \mapsto \mathbb{R}^+$, and a positive integer $k$
Output: a set $S$ of $k$ hole spanners
1: Construct a graph $G\prime = (V, E)$ with its information diffusion probability function $w\prime : E \mapsto \mathbb{R}^+$ from $G$, where
   $w\prime_{ij} = \log_2 \frac{1}{w_{ij}}$ for each edge $(v_i, v_j)$ in $G$;
2: Let set $S \leftarrow \varnothing$;
3: for $i \leftarrow 1$ to $k$ do
4:     Identify a user $u_i$ in $V \setminus S$ that blocks the maximum marginal number of information diffusion by invoking
       Algorithm 1;
5:     $S \leftarrow S \cup \{u_i\}$;
6: end for
7: return $S$.
```

4.2. Algorithm analysis

We now discuss the performance of Algorithm 2 as follows.

4.2.1. Analysis of the time complexity

We start by analyzing the time complexity of Algorithm 2.

Lemma 3. *Given a social network $G = (V, E)$, an information diffusion probability function $w : E \mapsto [0, 1]$, an information sharing rate function $f : V \mapsto \mathbb{R}^+$, and a positive integer k, the time complexity of Algorithm 2 is $O\left(knm + kn^2 \log n\right)$, where $n = |V|$ and $m = |E|$.*

Proof. We begin by analyzing the time complexity of Algorithm 1, since it is invoked k times by Algorithm 2. Consider the information diffused from a source node $s_j \in V$. The construction of a maximum likely diffusion tree T_j rooted at the source node s_j takes time $O(m + n \log n)$ by the invoking of Dijkstra's algorithm. It then takes $O(n)$ time to obtain tree $T\prime_j$ by removing the nodes in $D_j(S)$ from T_j. Finally, it takes $O(n)$ time to compute the marginal number $H\prime_j(v_l)$ of blocked information diffusion after removing each node v_l in tree $T\prime_j$ with a DFS starting from source s_j. Therefore, it takes time $O(m + n \log n) + O(n) + O(n) = O(m + n \log n)$ to compute the marginal number $H\prime_j(v_l)$ of blocked information diffusion after removing each node v_l in tree $T\prime_j$

Since Algorithm 1 computes the marginal number of blocked information diffusion in the n maximum likely diffusion trees after removing each node v_l, the time complexity of Algorithm 1 is $O(n(m + n \log n)) = O(nm + n^2 \log n)$.

Finally, since Algorithm 2 identifies a set S of k diversified hole spanners by invoking k times of Algorithm 1, the time complexity of Algorithm 2 is $O\left(knm + kn^2 \log n\right)$. The lemma then follows. □

4.2.2. Analysis of the approximation ratio

We now prove that the objective $H(S)$ of the considered problem in this paper is a nondecreasing submodular function. Then, following the study in [29], the greedy strategy in Algorithm 2 always deliver a $(1 - \frac{1}{e})$-approximate solution, where e is the base of the natural logarithm.

Theorem 1. *Given a social network $G = (V, E)$, an information diffusion probability function $w : E \mapsto [0, 1]$, an information sharing rate function $f : V \mapsto \mathbb{R}^+$, and a positive integer k, there is $(1 - \frac{1}{e})$-approximation algorithm, i.e., Algorithm 2, for the top-k diversified hole spanners problem with a time complexity of $O\left(knm + kn^2 \log n\right)$, where $n = |V|$ and $m = |E|$.*

Proof. We prove that function $H(\cdot)$ is a nondecreasing submodular function, where the value of $H(\cdot)$ is the number of blocked information diffusion after removing a set of hole spanners in network G.

First, following the definition of submodular functions (see Section 3.5), it is obvious that $H(\varnothing) = 0$. Then, for any two subsets A and B of V with $A \subseteq B$, it can be seen that, in the maximum likely diffusion tree T_j rooted at any source node s_j, any descendant v_l of hole spanners in set A in tree T_j also is a descendant of a spanner in set B, i.e., $D_j(A) \subseteq D_j(B)$, where $D_j(A)$ and $D_j(B)$ are the sets of descendants of spanners in sets A and B, respectively. Following Eq. (3) in Section 3.4, we have $H_j(A) = f_j \sum_{v_l \in D_j(A)} p_{jl} \leqslant f_j \sum_{v_l \in D_j(B)} p_{jl} = H_j(B)$. In addition, following Eq. (4) in Section 3.4, the number of blocked information diffusion $H(A)$ and $H(B)$ by hole spanners in sets A and B in network G are $H(A) = \sum_{j=1}^{n} H_j(A)$ and $H(B) = \sum_{j=1}^{n} H_j(B)$, respectively. Then, the value of $H(A)$ is no larger than the value of $H(B)$, i.e., $H(A) \leqslant H(B)$. Therefore, function $H(\cdot)$ is nondecreasing.

Finally, we show the submodularity of function $H(\cdot)$. Specifically, for any two subsets A and B of V with $A \subseteq B$ and any node v in $V \setminus B$, in the following, we prove that $H(A \cup \{v\}) - H(A) \geqslant H(B \cup \{v\}) - H(B)$.

We only show that function $H_j(\cdot)$ is submodular for each source $s_j \in V$ with $1 \leqslant j \leqslant n$. Since function $H(\cdot)$ is the sum of the n submodular functions $H_1(\cdot), H_2(\cdot), \ldots, H_n(\cdot)$, i.e., $H(\cdot) = \sum_{j=1}^{n} H_j(\cdot), H(\cdot)$ is submodular, too [29].

For any two subsets A and B of V with $A \subseteq B$, and any node v in $V \setminus B$, we distinguish our discussion into three cases: (i) v is a descendant of a node in set A, i.e., $v \in D_j(A)$; (ii) v is not a descendant of any node in set A, but a descendant of a node in set B, i.e., $v \notin D_j(A)$ and $v \in D_j(B)$; and (iii) v is not a descendant of any node in set B, i.e., $v \notin D_j(B)$.

We first consider Case (i) where $v \in D_j(A)$, e.g., v_9 in Fig. 4(a). Since v is a descendant of a node in set A and $A \subseteq B$, v must be a descendant of a node in set B, too, i.e., $v \in D_j(B)$. It can be seen that any descendant of v in T_j is also the descendant of nodes in sets A and B in T_j, see Fig. 4(a). Therefore, $D_j(A \cup \{v\}) = D_j(A)$ and $D_j(B \cup \{v\}) = D_j(B)$. Then, $H_j(A \cup \{v\}) - H_j(A) = f_j \sum_{v_l \in D_j(A \cup \{v\})} p_{jl} - f_j \sum_{v_l \in D_j(A)} p_{jl} = 0$ and $H_j(B \cup \{v\}) - H_j(B) = f_j \sum_{v_l \in D_j(B \cup \{v\})} p_{jl} - f_j \sum_{v_l \in D_j(B)} p_{jl} = 0$. We thus have

$$H_j(A \cup \{v\}) - H_j(A) = H_j(B \cup \{v\}) - H_j(B) = 0. \tag{8}$$

We then consider Case (ii) where $v \notin D_j(A)$ and $v \in D_j(B)$. That is, v is not a descendant of any node in set A (i.e., $v \notin D_j(A)$), but a descendant of a node in set B (i.e., $v \in D_j(B)$), e.g., v_{11} in Fig. 4(b). It can be seen that any descendant of v in T_j also is the descendant of a node in set B, but not the descendant of any node in set A in T_j, see Fig. 4(b). Then, $D_j(A) \subset D_j(A \cup \{v\})$ and $D_j(B \cup \{v\}) = D_j(B)$. Thus, we have $H_j(A \cup \{v\}) - H_j(A) = f_j \sum_{v_l \in D_j(A \cup \{v\})} p_{jl} - f_j \sum_{v_l \in D_j(A)} p_{jl} \geqslant 0$, while $H_j(B \cup \{v\}) - H_j(B) = f_j \sum_{v_l \in D_j(B \cup \{v\})} p_{jl} - f_j \sum_{v_l \in D_j(B)} p_{jl} = 0$. Therefore,

$$H_j(A \cup \{v\}) - H_j(A) \geqslant H_j(B \cup \{v\}) - H_j(B) = 0. \tag{9}$$

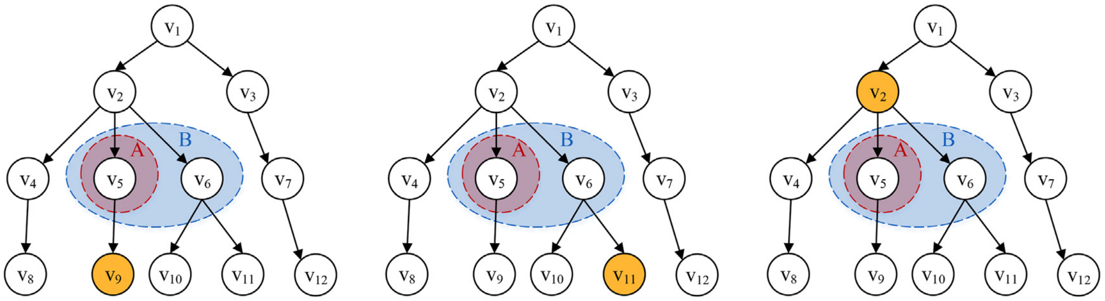
We finally study Case (iii) where $v \notin D_j(B)$. Since v is not a descendant of any node in set B and $A \subseteq B$, v also is not a descendant of any node in set A (i.e., $v \notin D_j(A)$), e.g., v_2 in Fig. 4(c). Following Eq. (3) in Section 3.4, $H_j(B \cup \{v\}) - H_j(B) = f_j \sum_{v_l \in D_j(B \cup \{v\})} p_{jl} - f_j \sum_{v_l \in D_j(B)} p_{jl} = f_j \sum_{v_l \in D_j(B \cup \{v\}) \setminus D_j(B)} p_{jl}$ and $H_j(A \cup \{v\}) - H_j(A) = f_j \sum_{v_l \in D_j(A \cup \{v\})} p_{jl} - f_j \sum_{v_l \in D_j(A)} p_{jl} = f_j \sum_{v_l \in D_j(A \cup \{v\}) \setminus D_j(A)} p_{jl}$. In order to prove that $H_j(A \cup \{v\}) - H_j(A) \geqslant H_j(B \cup \{v\}) - H_j(B)$, we show that any node in set $D_j(B \cup \{v\}) \setminus D_j(B)$ must be contained in set $D_j(A \cup \{v\}) \setminus D_j(A)$, i.e., $D_j(B \cup \{v\}) \setminus D_j(B) \subseteq D_j(A \cup \{v\}) \setminus D_j(A)$, which is shown as follows.

For any node v_l in set $D_j(B \cup \{v\}) \setminus D_j(B)$, v_l is a descendant of v, but not a descendant of any node in set B. Since $A \subseteq B$, v_l thus is not a descendant of any node in set A. Then, v_l is a descendant of v, but not a descendant of any node in set A. Therefore, v_l is contained in $D_j(A \cup \{v\}) \setminus D_j(A)$. We conclude that $D_j(B \cup \{v\}) \setminus D_j(B) \subseteq D_j(A \cup \{v\}) \setminus D_j(A)$. Then, we have

$$\begin{aligned} & H_j(A \cup \{v\}) - H_j(A) \\ = \; & f_j \sum_{v_l \in D_j(A \cup \{v\})} p_{jl} - f_j \sum_{v_l \in D_j(A)} p_{jl}, \text{by Eq.(3)} \\ = \; & f_j \sum_{v_l \in D_j(A \cup \{v\}) \setminus D_j(A)} p_{jl} \\ \geqslant \; & f_j \sum_{v_l \in D_j(B \cup \{v\}) \setminus D_j(B)} p_{jl}, \;\; \text{as} D_j(B \cup \{v\}) \setminus D_j(B) \subseteq D_j(A \cup \{v\}) \setminus D_j(A) \\ = \; & f_j \sum_{v_l \in D_j(B \cup \{v\})} p_{jl} - f_j \sum_{v_l \in D_j(B)} p_{jl} \\ = \; & H_j(B \cup \{v\}) - H_j(B). \end{aligned} \tag{10}$$

By combining InEq. (8), (9), (10), it can be seen that $H_j(A \cup \{v\}) - H_j(A) \geqslant H_j(B \cup \{v\}) - H_j(B)$. Then, the function $H_j(\cdot)$ is submodular. Since $H(\cdot)$ is the nonnegative linear combinations of the n functions $H_1(\cdot), H_2(\cdot), \ldots, H_n(\cdot), H(\cdot)$ is a submodular function, too.

Since $H(\cdot)$ is a nondecreasing submodular function, the greedy algorithm delivers a $(1 - \frac{1}{e})$-approximate solution [29]. The theorem then follows. □



(a) Case (i): $v_9 \in D_j(A)$.

(b) Case (ii): $v_{11} \notin D_j(A)$ and $v_{11} \in D_j(B)$

(c) Case (iii): $v_2 \notin D_j(B)$

Fig. 4. An illustration of the submodularity of function $H_j(\cdot)$.

5. Fast randomized algorithm

Although the proposed algorithm in the last section can deliver a $(1-\frac{1}{e})$-approximate solution, its running time $O\left(knm + kn^2 \log n\right)$ is not short (see Theorem 1), especially when there are millions of users in the network. To address the running time problem, in this section we propose a randomized algorithm with a much smaller time complexity.

5.1. Algorithm basic idea

A simple way to reduce the algorithm running time is that, we can estimate the number of blocked information diffusion by removing of a set S of hole spanners, by randomly selecting only L ($\leqslant n$) source nodes among the n source nodes in the social network, where the probabilities of selecting different nodes are identical (i.e., $\frac{L}{n}$), and L is a given positive integer, e.g., $L = 100\lceil \log_2 n \rceil$. Denote by $\overline{H_L(S)}$ the average number of blocked information diffusion from the L source nodes after removing nodes in S. The number of blocked information diffusion in network G thus can be estimated as $n \cdot \overline{H_L(S)}$, where n is the number of source nodes in the network. Then, the k nodes that block the maximum number of information diffusion from the L source nodes can be considered as the top-k diversified hole spanners.

We however find that the performance of the aforementioned uniform sampling is poor, since the average number $\overline{H_L(S)}$ of blocked information diffusion after removing a set S of hole spanners may converge slowly with the growth of the number L of selected source nodes. The rationale behind the slow convergence is that, the numbers of blocked information diffusion originated from different source nodes after removing the nodes in a set S may vary significantly. Then, the variance of the random variable $\overline{H_L(S)}$ may be large.

We now devise a fast randomized algorithm, which converges much faster than the way of uniform sampling. The key technique we use is called as *weighted sampling*, where different source nodes have different sampling probabilities. Specifically, denote by a_j the number of information diffusion originated from a source node s_j, where $1 \leqslant j \leqslant n$. Then, the sampling probability γ_j of source node s_j is proportional to the value of a_j. That is, given a positive integer L_s (e.g., $L_s = 100\lceil \log_2 n \rceil$), the sampling probability of a source node s_j is $\gamma_j = L_s \frac{a_j}{\sum_{i=1}^{n} a_i}$. Using the sampling probabilities γ_j, we can obtain a set V_s of sampled source nodes, and the expected number of sampled source nodes in V_s is $\sum_{j=1}^{n} \gamma_j = L_s \sum_{j=1}^{n} \frac{a_j}{\sum_{i=1}^{n} a_i} = L_s$.

In the development of the randomized algorithm, we need to address the following two questions.

(1) How to obtain the number a_j of information diffusion from each source node s_j?
(2) Given a set V_s of sampled source nodes, how to find the top-k diversified hole spanners?

We address the two questions one by one as follows.

5.2. Estimation of the number of information diffusion from each source node

A simple way of computing the number a_j of information diffusion originated from each source node s_j is to find the maximum likely diffusion tree T_j rooted at s_j, by invoking Procedure 1 n times, once for each node as the source. However, the time complexity of this way is as high as $O(n(m + n \log n))$, where the time complexity of Procedure 1 is $O(m + n \log n)$, n is the number of users and m is the number of links in social network G.

We here propose a randomized algorithm to estimate the number a_j of information diffusion originated from each source node s_j. Given a social network $G = (V, E)$, an information diffusion probability function $w : E \mapsto [0, 1]$, and an information sharing rate function $f : V \mapsto \mathbb{R}^+$, we first construct an graph $G^t = \left(V, E^t\right)$ with its information diffusion probability function $w^t : E \mapsto [0, 1]$ from G, where G^t is a transpose of G. That is, there is an edge $\left(v_i, v_j\right)$ in E^t if and only if edge $\left(v_j, v_i\right)$ is contained in E, and $w^t\left(v_i, v_j\right) = w\left(v_j, v_i\right)$. We can see that the information diffusion probability p^t_{ij} from a node v_i to another node s_j in graph G^t is identical to the information diffusion probability p_{ji} from s_j to v_i in graph G, i.e., $p^t_{ij} = p_{ji}$.

We then estimate the number a_j of information diffusion from each source node s_j as follows. Given a positive integer L_a (e.g., $L_a = 200\lceil \log_2 n \rceil$), we randomly select a set V_a of L_a nodes from V, where the probabilities of selecting different nodes are identical. For each node $v_i \in V_a$, we find the maximum likely diffusion tree T_i rooted at v_i in graph G^t, by invoking Procedure 1. For each source node s_j in tree T_i, denote by p^t_{ij} the information diffusion probability from node v_i to node s_j in tree T_i. Then, we estimate the value of a_j as $\hat{a}_j = \frac{\sum_{i=1}^{L_a} f_j \cdot p^t_{ij}}{L_a} \cdot n$, where f_j is the information sharing rate of node s_j. The detailed algorithm is described in Algorithm 3. It can be seen that the time complexity of Algorithm 3 is only $O(L_a(m + n \log n))$.

```
Algorithm 3: Estimate the number of information diffusion originated from each source node
Input: A social network $G = (V, E)$, an information diffusion probability function $w : E \mapsto \mathbb{R}^+$, an information sharing rate
   function $f : V \mapsto \mathbb{R}^+$, and a positive integer $L_a$
Output: Estimate the number of information diffusion originated from each source node
Construct a transpose graph $G^t = \left(V, E^t\right)$ with its information diffusion probability function $w^t : E \mapsto [0, 1])$ from $G$;
Randomly select a set $V_a$ of $L_a$ nodes from $V$, where the probabilities of selecting different nodes are identical.
For each source node $s_j \in V$, let $\hat{a}_j \leftarrow 0$;/* the estimated number of information diffusion from source node $s_j$ */
for each node $v_i$ in $V_a$ do
    Find the maximum likely diffusion tree $T_i$ rooted at node $v_i$ in graph $G^t$, by invoking Procedure 1 in Section 3.3;
    Calculate the information diffusion probability $p^t_{ij}$ from node $v_i$ to each other source node $s_j \in T_j$;
    for each node $s_j \in T_i$ do
        $\hat{a}_j \leftarrow \hat{a}_j + f_j \cdot p^t_{ij}$;
    end for
end for
Let $\hat{a}_j \leftarrow \frac{\hat{a}_j}{L_a} \cdot n$ for each source node $s_j \in V$;
return the estimated number $\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_n$ of information diffusion originated from source nodes $s_1, s_2, \ldots, s_n$,
   respectively.
```

5.3. Identify the top-k diversified hole spanners

Having the estimated number $\hat{a}_j$ of information diffusion from each source node s_j, given a positive integer L_s (e.g., $L_s = 100\lceil \log_2 n \rceil$), the sampling probability of source node s_j is set $\gamma_j = L_s \frac{\hat{a}_j}{\sum_{i=1}^{n} \hat{a}_i}$. Using the sampling probabilities γ_j, we obtain a set V_s of sampled source nodes, and the expected number of sampled source nodes in V_s is $\sum_{j=1}^{n} \gamma_j = L_s$.

The basic idea of the randomized algorithm for identifying the top-k diversified hole spanners is similar to Algorithm 2, i.e., using a greedy strategy. Specifically, let S be the set of found hole spanners by the algorithm. Initially, $S = \emptyset$. Within the ith iteration ($1 \leqslant i \leqslant k$), the algorithm identifies a node u_i in set $V \setminus S$ that blocks the maximum marginal number of information diffusion, i.e., $u_i = \arg\max_{v \in V \setminus S} \{H(S \cup \{v\}) - H(S)\}$. The procedure continues until set S contains k nodes.

Different from Algorithm 2, in each iteration i ($1 \leqslant i \leqslant k$), the randomized algorithm estimates, rather than precisely computes, the marginal number of blocked information diffusion after removing each node in $V \setminus S$. Denote by $H\prime(v)$ the marginal number of blocked information diffusion after removing a node v in $V \setminus S$. We estimate the value of $H\prime(v)$ as

$$\overline{H\prime(v)} = \sum_{s_j \in V_s} \frac{H\prime_j(v)}{\gamma_j},$$

where $H\prime_j(v)$ is the marginal number of blocked information diffusion after removing node v in the maximum likely diffusion tree T_j rooted at source node s_j, and γ_j is the sampling probability of node s_j. We later show that $\overline{H\prime(v)}$ is an unbiased estimator of $H\prime(v)$. The randomized algorithm is described in Algorithm 4.

Algorithm 4: Randomized algorithm for the problem (DHSFast).

```
Input: A social network $G = (V, E)$, an information diffusion probability function $w : E \mapsto [0, 1]$, an information sharing
  rate function $f : V \mapsto \mathbb{R}^+$, a positive integer $k$, a sample size $L_a$ for estimating the number of information diffusion from
  each node, and an expected sample size $L_s$ for identifying the top-$k$ hole spanners
Output: a set $S$ of $k$ hole spanners
Construct a graph $G' = (V, E)$ with its information diffusion probability function $w' : E \mapsto \mathbb{R}^+$ from $G$, where
  $w'_{ij} = \log_2 \frac{1}{w_{ij}}$ for each edge $(v_i, v_j)$ in $G$;
Estimate the number $\hat{a}_j$ of information diffusion originated from each source node $s_j$ in $V$, by invoking Algorithm 3;
For each source node $s_j$ in $V$, let $\gamma_j \leftarrow L_s \frac{\hat{a}_j}{\sum_{i=1}^{n} \hat{a}_i}$;/* the sampling probability of source node $s_j$ */
Obtain a set $V_s$ of nodes from $V$, by sampling each source node $s_j$ in $V$ with a probability of $\gamma_j$;
Let $S \leftarrow \emptyset$;/* the set of identified hole spanners */
for $i \leftarrow 1$ to $k$ do
  /* Identify a user $u_i$ that blocks the maximum marginal number of information diffusion */
  For each $v_l \in V$, let $\overline{H'(v_l)} \leftarrow 0$; /* the estimated marginal number of blocked information diffusion by each node $v_l$
  */
  for each source node $s_j$ in $V_s$ do
    Find the maximum likely diffusion tree $T_j$ in $G$ rooted at source node $s_j$, by invoking Procedure 1 in Section 3.3;
    Obtain a tree $T'_j$ by the removals of the descendants of nodes in $S$, i.e., nodes in $D_j(S)$, from $T_j$;
    Calculate the marginal number $H'_j(v_l)$ of blocked information diffusion of each node $v_l \in T'_j$ by performing a
  DFS starting from the source node $s_j$;
    for each node $v_l \in T'_j$ do
      $\overline{H'(v_l)} \leftarrow \overline{H'(v_l)} + \frac{H'_j(v_l)}{\gamma_j}$;
    end for
  end for
  Assume that $u_i$ is a node in $V$ with $\overline{H'(u_i)} = \max_{v_l \in V \setminus S} \left\{ \overline{H'(v_l)} \right\}$.
  Let $S \leftarrow S \cup \{u_i\}$;
end for
return $S$.
```

5.4. Algorithm analysis

Lemma 4. *Denote by $H'(v)$ the marginal number of blocked information diffusion after removing a node v in $V \setminus S$. The value of $H'(v)$ is estimated as $\overline{H'(v)} = \sum_{s_j \in V_s} \frac{H'_j(v)}{\gamma_j}$. Then, $\overline{H'(v)}$ is an unbiased estimator of $H'(v)$.*

Proof. Denote by $H'_j(v)$ the marginal number of blocked information diffusion after removing a node v in the maximum likely diffusion tree T_j rooted at a source node s_j in V. Then, $H'(v) = \sum_{s_j \in V} H'_j(v)$. The expected value of random variable $\overline{H'(v)}$ is

$$\mathbb{E}\left[\overline{H'(v)}\right] = \sum_{s_j \in V} \left(\gamma_j \cdot \frac{H'_j(v)}{\gamma_j} + \left(1 - \gamma_j\right) \cdot 0 \right) = \sum_{s_j \in V} H'_j(v) = H'(v). \tag{12}$$

The lemma then follows. □

Theorem 2. *Given a network $G = (V, E)$, an information diffusion probability function $w : E \mapsto [0, 1]$, and an information sharing rate function $f : V \mapsto \mathbb{R}^+$, a sample size L_a for estimating the number of information diffusion from each node, and an expected sample size L_s for identifying the top-k hole spanners, there is a randomized algorithm, i.e., Algorithm 4, applied to the top-k diversified hole spanners problem, and its expected time complexity is $O((L_a + kL_s)(m + n \log n))$, where $n = |V|$ and $m = |E|$.*

Proof. In the following, we analyze the expected time complexity of Algorithm 4. It takes $O(L_a(m + n \log n))$ time to estimate the number $\hat{a}_j$ of information diffusion originated from each source node s_j in V, by invoking Algorithm 3, since Dijkstra's algorithm is invoked L_a times. Algorithm 4 then identifies the top-k hole spanners with a greedy strategy. Within the ith iteration, the algorithm identifies a node u_i that blocks the maximum marginal number of information diffusion, which takes time $O(L_s(m + n\log n))$ on average, since the expected number of nodes in V_s is L_s. Therefore, the expected time complexity of Algorithm 4 is $O(L_a(m + n\log n)) + k \cdot O(L_s(m + n\log n)) = O((L_a + kL_s)(m + n\log n))$. □

6. Performance evaluation

We evaluate the algorithm performance with six real-world network datasets.

6.1. Experimental environment

We adopt six real social network datasets from different fields to study the algorithm performance for the top-k diversified hole spanner problem, see Table 1. The first dataset `email-Eu-core` is a network of e-mails among members in a European research institution, and ground-truth community memberships of users are also known. The second dataset `wiki-Vote` is a voting network for the election of Wikipedia administrators, and was collected from the inception of Wikipedia in 2008. The third dataset `cit-HepPh` is a citation network of high energy physics phenomenology category for a period of 124 months. The fourth dataset is a snapshot of the online social network `Facebook`. The fifth dataset `email-EuAll` is a network of e-mails and was generated from an European research institution for a period of 18 months. The final dataset `wiki-Talk` is a communication network consisting of talk pages, where users share and discuss updates of various articles on Wikipedia. The `Facebook` dataset is obtained from [39], and the other five datasets are acquired from the SNAP repository [23].

Following existing studies [2,27,35], a small number of users are very active in sharing their information in a social network, while most of the users are inactive. The famous "90-9-1" rule says that 90% users are only observe and/or read information but never create, 9% users create a little information, while only 1% users actively contribute new information [2,27,35]. Then, for each of the six social networks, we partition the users in the network into three groups and the numbers of users in the three groups accounts for 90%, 9%, and 1% of the total number of users, respectively, based on the number of neighbors of each user, i.e., node degrees, and a user is more likely to be an active user if its degree is larger. Specifically, a user v_i has a probability of $0.01n \cdot \frac{d_i}{\sum_{j=1}^{n} d_j}$ to be chosen as one of the active users that accounts for 1% of total users, where n is the number of users in the network, d_i and d_j are the degrees of users v_i and v_j, respectively. After having chosen the 1% active users, we then partition the rest 99% users into two groups, which accounts for 9% and 90% users in the network, by using the similar way. The information sharing rates f_i of users v_i in the three groups are randomly selected from the three intervals [0.011–0.078], [0.238–0.374], and [0.59–0.75], respectively [27]. The information diffusion probability w_{ij} of each edge in a social network is randomly chosen from the interval $[0, 0.1]$ [39]. The number k of diversified hole spanners is from 1 to 50. The values of parameters are summarized in Table 2.

Benchmark algorithms: To study the effectiveness and efficiency of the approximation algorithm `DHS` and randomized algorithm `DHSFast` proposed in this paper, we consider eight benchmark algorithms. The first five benchmark algorithms identify redundant hole spanners, while the later three algorithms find diversified, i.e., nonredundant, hole spanners.

Table 1
Statistics of six real-world networks.

Datasets	# of Nodes	# of Edges	Avg. degree	Diameter	Avg. clc[1]
email-Eu-core	1,005	25,571	25.4	7	0.3994
wiki-Vote	7,115	103,689	14.6	7	0.1409
cit-HepPh	34,546	421,578	12.2	12	0.2848
Facebook	63,731	1,634,180	25.6	15	0.1477
email-EuAll	265,214	420,045	1.6	14	0.0671
wiki-Talk	2,394,385	5,021,410	2.1	9	0.0526

1: clc: clustering coefficient.

Table 2
Parameters and values in the experiments.

Parameter	Value
Information sharing rate f_i	90% users: [0.011–0.078] 9% users: [0.238–0.374] 1% users: [0.59–0.75]
Diffusion probability w_{ij} of each edge	[0, 0.1]
Number of hole spanners k	[1, 50]
Sample size L_a in Algorithm `DHSFast`	$200\lceil \log_2 n \rceil$
Expected sample size L_s in Algorithm `DHSFast`	$100\lceil \log_2 n \rceil$

We first briefly introduce the first five benchmark algorithms that identify top-k *redundant* hole spanners. That is, they first measure each node with a score for acting as a hole spanner. They then identify the k nodes with the maximum or minimum scores as the top-k hole spanners. We introduce the five algorithms one by one as follows.

(1) Algorithm `Constraint` [5] calculates the constraint score of each node by its neighbors, and chooses the k nodes with the lowest scores.
(2) Algorithm `PathCount` [17] first measures each node by the number of shortest paths passing through it, then selects the top-k nodes.
(3) Algorithm `HIS` [25] first measures each node by the probability that the node connects with opinion leaders in multiple communities, then selects the top-k nodes, where communities are given.
(4) Algorithm `HAM` [20] selects the top-k nodes whose neighbors belong to as many communities as possible.
(5) Algorithm `maxBlockFast` [39] first estimates the number of blocked information diffusion after removing each node, then selects the top-k nodes.
We then introduce the three algorithms that identify *diversified* hole spanners, which means that the communities spanned by the found hole spanners are highly nonredundant.
(6) Algorithm `maxCoverage` [21] identifies k nodes, so that the number of different communities with which the k nodes connect is maximized.
(7) Algorithm `MaxD` [25] finds k nodes, so that their removals result in the maximum decrease of the minimal cut in a network.
(8) Algorithm `APGreedy` [40] identifies k nodes to maximize the communication cost in the network after removing the k nodes.

Notice that the time complexities of the ten algorithms are listed in Table 3.

Since there is a prerequisite of applying algorithms `HIS`, `MaxD` and `maxCoverage`, i.e., communities in the networks must be given in advance, we use an existing community detection algorithm [3] to find communities in networks `wiki-Vote`, `cit-HepPh`, `Facebook`, `email-EuAll`, and `wiki-Talk`, as ground-truth communities in them are unknown. On the other hand, ground-truth communities in network `email-Eu-core` are known.

Notice that, due to the high space complexity $O(n^2)$ of algorithm `HAM` [20,39], its performance is compared only in the smallest three networks, i.e., the `email-Eu-core` network with about one thousand nodes, the `wiki-Vote` network with about seven thousand nodes, and the `cit-HepPh` network with about 34 thousand nodes, where the memory consumptions of algorithm `HAM` in the three networks are about 25 MB, 1.2 GB, 30 GB, respectively. In contrast, the memory consumption of algorithm `HAM` in the fourth smallest network `Facebook` with about 63 thousand nodes is expected to grow to more than 90 GB, which is larger than the memory capacity 64 GB of our server.

All algorithm codes are written by the programming language C++, and they are run on a server, which contains an Intel (R) Xeon(R) CPU E5-2680 v4 with 2.4 GHz and 28 threads, and a 64 GB RAM. Notice that we implement the algorithms in parallel by adopting multiple threads.

6.2. Algorithm performance

In this paper, we consider the prevention of the widespread of misinformation. Therefore, we evaluate the effectiveness of the top-k hole spanners identified by an algorithm in terms of the numbers of blocked information diffusion, see Eq. (3) and Eq. (4) in Section 3.4 for its definition. We also compare the running times of different algorithms.

Table 3
Time complexities of different algorithms.

Algorithms	Time complexities
DHS	$O(kn(m + n\log n))$
DHSFast	$O(100k\log n(m + n\log n))$
Constraint	$O\left(n + m + nd_{max}^2\right)$
PathCount	$O(nm + n^2)$
HIS	$O\left(2^{\gamma\cdot\min\{c,c_{max}\}}m\right)$
HAM	$O(n^3)$
maxBlockFast	$O(n\log n(m + n))$
maxCoverage	$O(ckn)$
MaxD	$O\left(2^{\gamma\cdot\min\{c,c_{max}\}}nm\right)$
APGreedy	$O(k(m + n))$

1 d_{max}: the maximum degree of nodes in a network.
2 γ: a parameter depends on the network with $0 < \gamma \leqslant 1$.
3 c: the number of communities in a network.
4 c_{max}: a predefined constant, e.g., 64.

6.2.1. Blocked information diffusion

We first evaluate the algorithm effectiveness in the `email-Eu-core` network. Fig. 5(a) shows the number of blocked information diffusion after removing the found hole spanners in each of the ten algorithms increases with the growth of k. It also can be seen that both algorithms `DHS` and `DHSFast` clearly outperform the eight benchmark algorithms. Specifically, the numbers of blocked information diffusion in the proposed algorithms `DHS` and `DHSFast` are almost identical when $k = 50$, and the number of blocked information diffusion after removing the hole spanners found by algorithm `DHS` is about 6% ($\approx \frac{670-632}{632}$) and 15% larger than those in the best two benchmark algorithms `MaxD` and `maxCoverage`, respectively, where the numbers of blocked information diffusion after removing the hole spanners found by algorithms `DHS`, `DHSFast`, `Constraint`, `PathCount`, `HIS`, `HAM`, `maxBlockFast`, `maxCoverage`, `MaxD`, `AP-Greedy` are 670.5, 670, 207, 568, 327, 150, 434, 583, 632, and 374, respectively, when $k = 50$.

Fig. 5(b) plots the numbers of communities spanned by the top-k hole spanners found by different algorithms in the `email-Eu-core` network. It is clear that the hole spanners found by algorithm `maxCoverage` bridge the maximum number of communities among the ten algorithms, since the objective of algorithm `maxCoverage` is to bridge the maximum number of communities, while the objectives of other algorithms are not. However, it can be see from Fig. 5(a) that the nodes identified by algorithm `maxCoverage` are not the best hole spanners to block the misinformation in the network, since it considered only the structure of the network, but ignored the tie strengths between the found hole spanners and their bridged communities. On the other hand, Fig. 5(b) shows that the numbers of spanned communities by the proposed algorithms `DHS` and `DHSFast` are about 97% of algorithm `maxCoverage`.

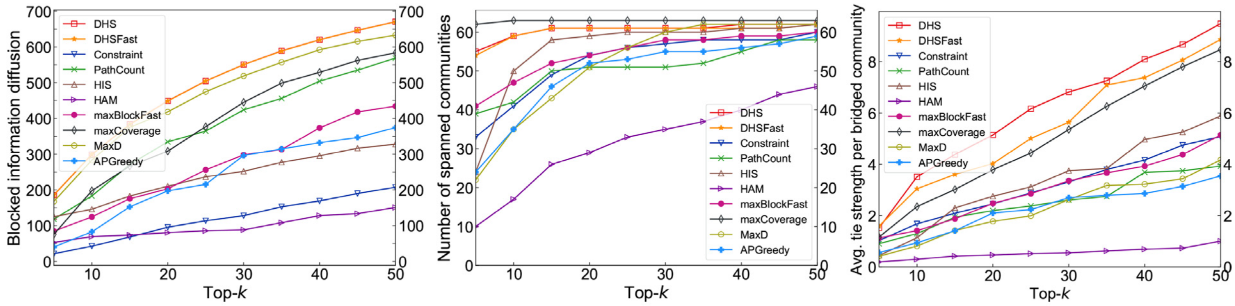
We further show that the hole spanners found by algorithms `DHS` and `DHSFast` have stronger tie strengths than those by other algorithms. Following the work of Burt [8], to obtain a higher profit, the tie strength between the identified hole spanners and each community they span should be large. We define the average tie strength per bridged community as follows. Given a social network $G = (V, E)$, assume that $S = \{u_1, u_1, \ldots, u_k\}$ is the set of top-k hole spanners found by an algorithm and $\mathscr{C} = \{C_1, C_2, \ldots, C_M\}$ is the set of nonredundant communities spanned by the nodes in S, where M is the number of bridged communities. Then, *the average tie strength per bridged community* by the nodes in S is

$$\rho(S) = \frac{\sum\limits_{u_i \in S, v_j \in \cup_{l=1}^{M} C_l} w(u_i, v_j)}{M}, \tag{13}$$

where $w(u_i, v_j)$ is the tie strength between users u_i and v_j. It can be seen from Fig. 5(c) that the average tie strengths per bridged community by algorithms `DHS` and `DHSFast` are least 12% and 5% larger than those by the other eight algorithms, because the hole spanners identified by the both algorithms have many friends in their bridged communities, or they interact frequently with these friends, i.e., they establish strong connections with their connected communities. In contrast, although the hole spanners identified by algorithm `maxCoverage` connect with more communities, the tie strength between the hole spanners and their bridged communities are weaker.

Therefore, it can be seen that the hole spanners found by both algorithms `DHS` and `DHSFast` not only bridge many communities (see Fig. 5(b)), but also have strong ties with their bridged communities (see Fig. 5(c)), thereby the removals of the hole spanners block more information propagations (see Fig. 5(a)).

We then study the performance of different algorithms in other five networks `wiki-Vote`, `cit-HepPh`, `Facebook`, `email-EuAll` and `wiki-Talk`, in terms of the blocked information diffusion by found hole spanners. It can be seen from Fig. 6 that the hole spanners identified by algorithms `DHS` and `DHSFast` are much better than those detected by other benchmark algorithms, especially in the `Cit-HepPh` and `wiki-Talk` networks. For example, Fig. 6(b) shows that after removing the top-50 nodes found by algorithms `DHS` and `DHSFast` in the `Cit-HepPh` network, the numbers of blocked information

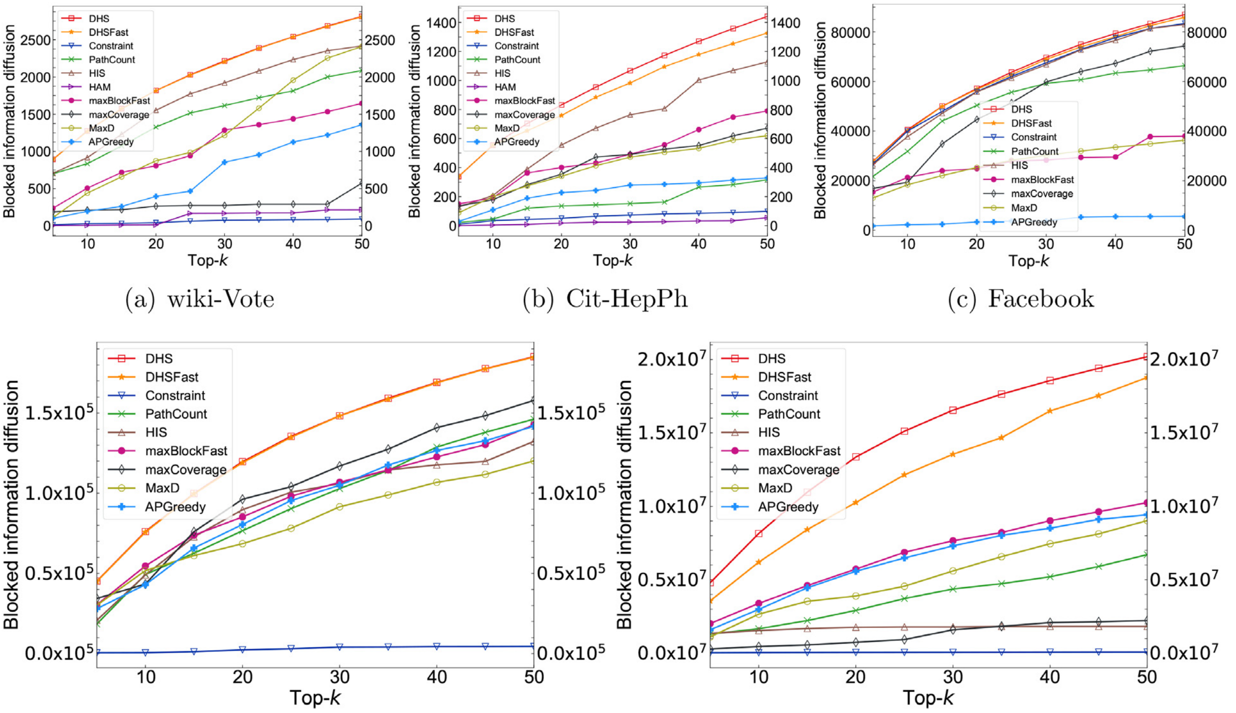


(a) Blocked information diffusion

(b) Number of spanned communities

(c) Average tie strength per bridged community

Fig. 5. Algorithms performance of different algorithms in the `email-Eu-core` network.



(a) wiki-Vote (b) Cit-HepPh (c) Facebook

(d) email-EuAll (e) wiki-Talk

Fig. 6. Blocked information diffusion by different algorithms in various networks.

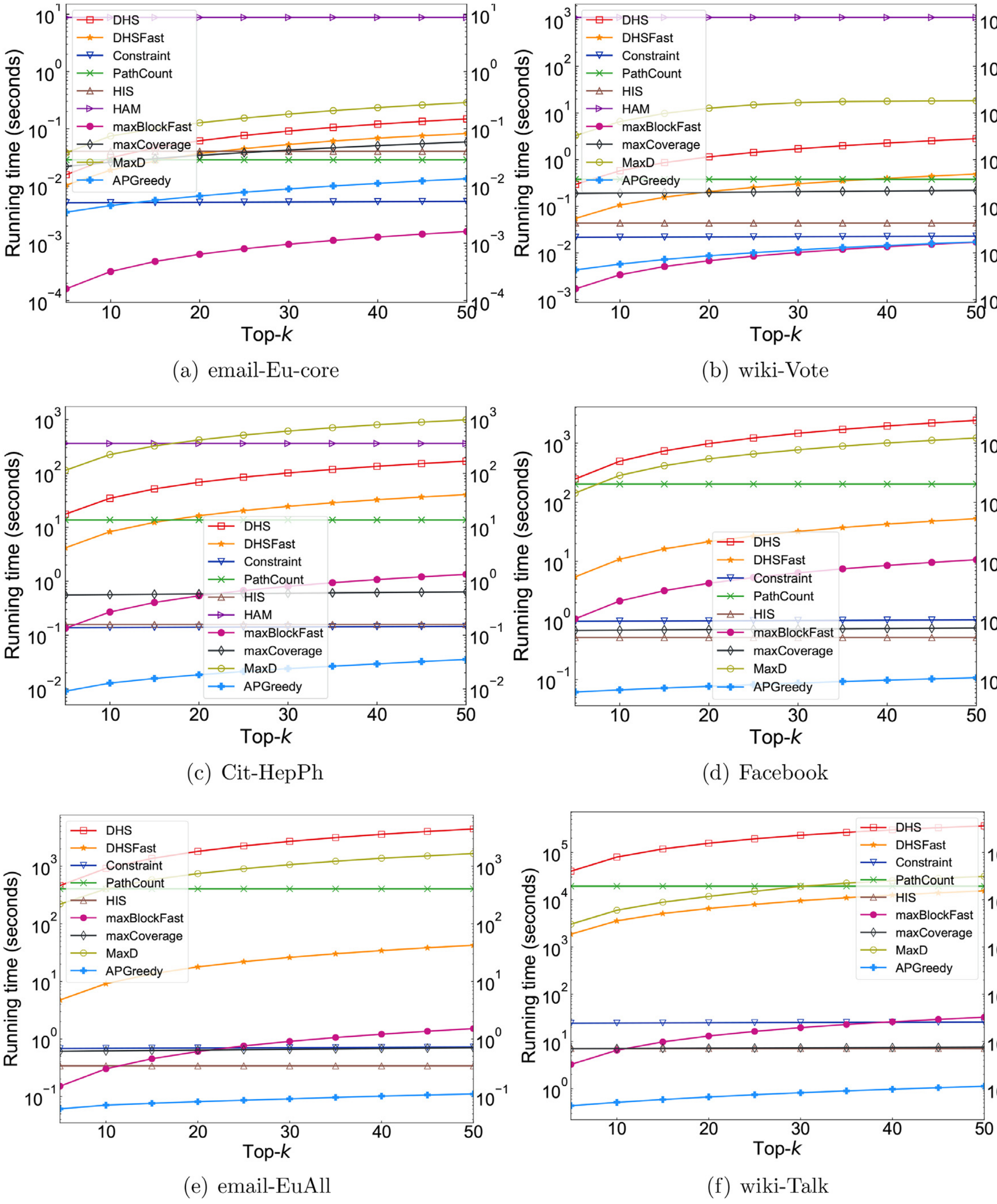
diffusion are about 28% $\left(\approx \frac{1{,}441-1{,}128}{1{,}128}\right)$ and 18% $\left(\approx \frac{1{,}327-1{,}128}{1{,}128}\right)$ larger than the best existing algorithm HIS among the eight existing algorithms, respectively. In addition, Fig. 6(e) plots that after removing the top-50 nodes found by algorithms DHS and DHSFast in the wiki-Talk network, the numbers of blocked information diffusion are 96% $(\approx \frac{20{,}184{,}828-10{,}252{,}709}{10{,}252{,}709})$ and 83% larger than the best algorithm maxCoverage. Notice that we here do not show the performance of different algorithms in terms of the numbers of spanned communities and average tie strengths per bridged community in the five networks, due to excessive long space of this paper. It must be mentioned that the proposed algorithms DHS and DHSFast show similar performance in the five network as them in the email-Eu-core network (see Fig. 5(b) and (c)).

In summary, it can be seen that the blocked information diffusions by the top-*k* hole spanners found by the proposed algorithms DHS and DHSFast are larger than those by the other benchmark algorithms in the six social networks. In addition, the experiment results validate our claim in Section 1.2 that the top-*k* hole spanners identified by the proposed algorithms DHS and DHSFast not only connect to multiple nonredundant communities, but also establish strong connections with their bridged communities.

6.2.2. Algorithm running time

Fig. 7 plots the running times of different algorithms in different networks. It can be seen that in small and medium-scale networks, the running times of algorithms MaxD and HAM are much longer than other algorithms. Specifically, Fig. 7(a) demonstrates that algorithm HAM has the longest running time (about 9 s), algorithm DHS takes 0.14 s to find the top-50 hole spanners, algorithm MaxD takes 0.28 s, whereas the proposed randomized algorithm DHSFast takes only about 0.08 s. Furthermore, the other algorithms take less than 0.1 s to find the top-*k* hole spanners. Fig. 7(b) and (c) show that the running time of algorithm DHS is only 0.2% and 17% of that of the slowest algorithm, respectively, when the number of hole spanners *k* is 50. In contrast, to identify the top-50 hole spanners, the running times of algorithm DHSFast are less than 0.5 and 40 s, respectively, which are about $\frac{1}{6}\left(=\frac{0.5}{3}\right)$ and $\frac{1}{4}\left(\approx \frac{40}{170}\right)$ of the running time of algorithm DHS, respectively. This implies that the proposed randomized algorithm DHSFast can not only find good hole spanners, but also enjoy short running time.

We further investigate the algorithm efficiency in the three larger networks Facebook, email-EuAll and wiki-Talk, which are plotted in Fig. 7(d), (e), and (f), respectively. Notice that we did not compare with algorithm HAM in the three networks, due to its high space complexity $O(n^2)$. It can bee seen that with the growth of the network scale, the randomized algorithm DHSFast with a time complexity of $O(k \log n(m + n \log n))$ is much more efficient than algorithm DHS with a time complexity of $O(kn(m + n \log n))$. For example, in the Facebook network, it takes 40 min and 53 s for algorithms DHS and



(a) email-Eu-core

(b) wiki-Vote

(c) Cit-HepPh

(d) Facebook

(e) email-EuAll

(f) wiki-Talk

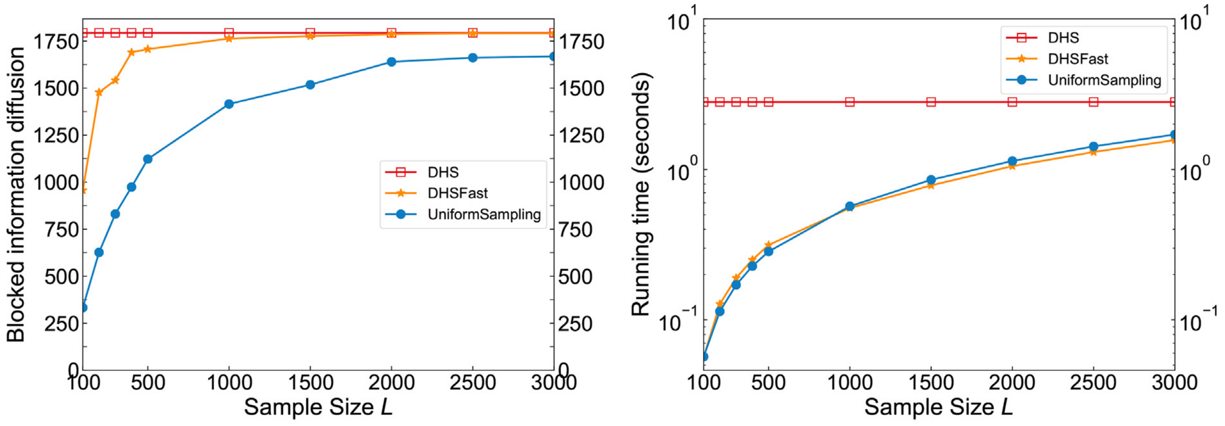
Fig. 7. Algorithm running times of different algorithms in various networks.

DHSFast to identify the top-50 hole spanners, and it takes approximately 4.2 days and 4.2 h, respectively, in the wiki-Talk network.

6.3. The convergence analysis of the randomized algorithm

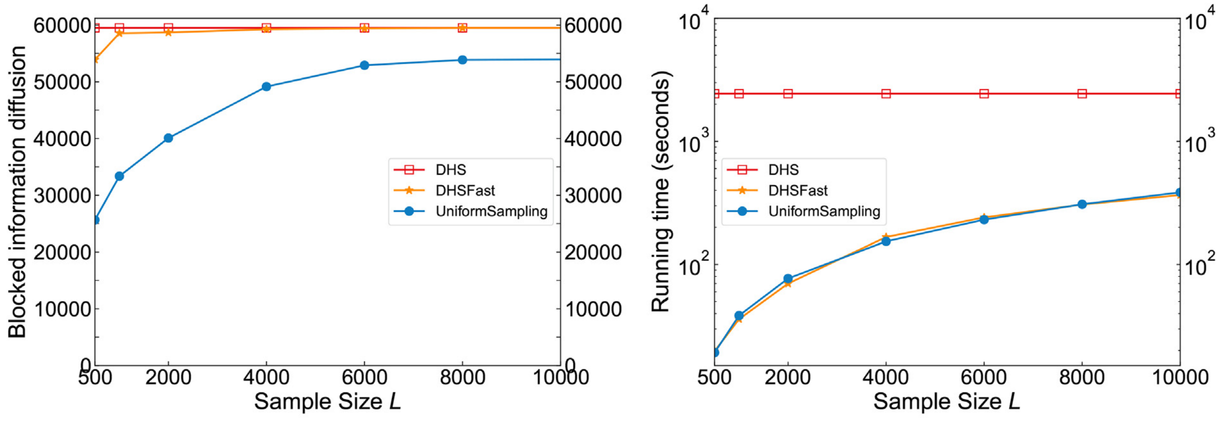
We now discuss the convergence performance of the randomized algorithm DHSFast, which uses a weighted sampling technique, see Section 5.3. We consider a benchmark algorithm UniformSampling. The basic idea of algorithm UniformSampling is described as follows. Given a positive integer L, it first randomly selects L source nodes from the n nodes in set V (the sampling probabilities of different nodes are identical), then identifies the k nodes and remove them from the network G, such that the number of blocked information diffusion from the L selected source nodes is maximized, by adopting the similar greedy strategy in algorithm DHS. The time complexity of algorithm UniformSampling is $O(kL(m + n \log n))$, where n is the number of users and m is the number of links in the considered network. To fairly compare the convergence performance between algorithms DHSFast and UniformSampling, in algorithm DHSFast, the sample size L_a for estimating the number of information diffusion from each source node and the expected sample size L_s for identifying the top-k hole spanners are set as $L_a = 2L$ and $L_s = \frac{k-2}{k}L$, respectively. Following Theorem 2, the expected time complexity of algorithm DHSFast then is $O((L_a + kL_s)(m + n \log n)) = O\big((2L + k\frac{k-2}{k}L)(m + n \log n)\big) = O(kL(m + n \log n))$. Therefore, algorithms DHSFast and UniformSampling almost have the same time complexity.

Fig. 8(a) shows the convergence performance of algorithms DHSFast and UniformSampling in the wiki-Vote network with about seven thousand nodes and 100 thousand edges, by varying the sample size L from 100 to 3,000. Fig. 8(a) shows that, after removing the nodes found by algorithm DHSFast, the number of information blocked diffusions increases very fast, when the sample size L increases from 100 to 500, but only slightly grows when L is larger than 500. In contrast, the number of blocked information diffusion after removing the nodes found by algorithm UniformSampling increases much slower when L increases from 100 to 500. For example, when the value of L increases from 100 to 500, the number of blocked information diffusion in algorithm DHSFast grows from 53% to as high as 95% of that by algorithm DHS, while the number of



(a) blocked information diffusion (b) Algorithm running time

Fig. 8. The convergence performance of algorithms DHSFast and UniformSampling with the growth of sample size L in the wiki-Vote network with about seven thousand nodes and 100 thousand edges.



(a) blocked information diffusion (b) Algorithm running time

Fig. 9. The convergence performance of algorithms DHSFast and UniformSampling with the growth of sample size L in the Facebook network with about 63 thousand nodes and 1.6 million edges.

blocked information diffusion in algorithm UniformSampling increases from 18% to only 62% of that by algorithm DHS. Fig. 8(b) plots that algorithms DHSFast and UniformSampling almost have identical running times.

We also evaluate the convergence performance of algorithms DHSFast and UniformSampling in the Facebook network with about 63 thousand nodes and 1.6 million edges, by varying the sample size L from 500 to 10,000. Fig. 9(a) demonstrates the number of blocked information diffusion after removing the nodes found by algorithm DHSFast increases to as high as 98% of that by algorithm DHS when $L = 1,000$, whereas the number of blocked information diffusion in algorithm UniformSampling is only about 56% of that by algorithm DHS. This indicates that the proposed randomized algorithm DHSFast converges very fast when L grows. In addition, Fig. 9(b) demonstrates that the gap between the running times of algorithms DHSFast and UniformSampling is very small.

7. Conclusion

In this paper, we studied the problem of top-k diversified hole spanners identification in social networks to block the maximum number of information diffusion, where not only the identified hole spanners connect nonredundant communities, but also we consider the tie strengths between different pairs of users and the different information sharing rates of different users. We devised a novel $(1-\frac{1}{e})$-approximation algorithm with a time complexity of $O(kn(m+n\log n))$ for the problem, where n is the number of users and m is the number of links in the considered social network. We also developed a fast randomized algorithm with a much smaller time complexity. We finally evaluated the algorithm performance. Our experiment results demonstrated that the numbers of blocked information diffusion after removing the found nodes in the proposed algorithms are up to 80% larger than those in existing algorithms. In addition, the number of blocked information diffusion by the randomized algorithm is only slightly smaller than that in the approximation algorithm, whereas the randomized algorithm is from 1.75 to 100 times faster than the approximation algorithm.

CRediT authorship contribution statement

Mengshi Li: Software, Validation, Data curation, Writing – original draft, Visualization. **Jian Peng:** Conceptualization, Methodology, Resources, Writing – review & editing, Funding acquisition. **Shenggen Ju:** Conceptualization, Resources, Project administration. **Quanhui Liu:** Conceptualization, Methodology, Resources. **Hongyou Li:** Conceptualization, Resources. **Weifa Liang:** Conceptualization, Formal analysis. **Jeffrey Xu Yu:** Conceptualization, Writing – review & editing. **Wenzheng Xu:** Conceptualization, Formal analysis, Investigation, Writing – review & editing, Supervision.

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.

Acknowledgements

The work by Wenzheng Xu was supported by the National Natural Science Foundation of China with Grant No. 61602330. The work by Jian Peng was supported by the National Key Research and Development Program of China with Grant No. 2017YFB0202403, and Sichuan Science and Technology Program with Grant Nos. 2018GZ0098, 2017GZDZX0003, and 2020YFG0089. The work by Hongyou Li was supported by Sichuan Science and Technology Program with Grant No. 2020YFG0304.

Appendix A. Proof of Lemma 1

Proof. Suppose that $P_{st} = \left\langle s, v_1, v_2, \ldots, v_{n_j}, t \right\rangle$ is not the path in G from s to t with the maximum diffusion probability. Let $P^*_{st} = \left\langle s, v^*_1, v^*_2, \ldots, v^*_{n^*_j}, t \right\rangle$ be the path in G from s to t with the maximum diffusion probability.

Denote by p_{st} and p^*_{st} the information diffusion probabilities of paths P_{st} and P^*_{st} respectively, which can be calculated by Eq. (1). Then, $p_{st} < p^*_{st}$.

On the other hand, we claim that the distance d^*_{st} of path P^*_{st} is strictly less than the distance d_{st} of path P_{st} in $G\prime$, since

$$\begin{aligned} d^*_{st} &= \sum_{i=0}^{n^*_j} w\prime(v^*_i, v^*_{i+1}) \\ &= \sum_{i=0}^{n^*_j} \log_2 \tfrac{1}{w(v^*_i, v^*_{i+1})}, \ \mathrm{as} w\prime(v^*_i, v^*_{i+1}) = \log_2 \tfrac{1}{w(v^*_i, v^*_{i+1})} \\ &= \log_2 \prod_{i=0}^{n^*_j} \tfrac{1}{w(v^*_i, v^*_{i+1})} \\ &= \log_2 \frac{1}{\prod_{i=0}^{n^*_j} w(v^*_i, v^*_{i+1})} \\ &= \log_2 \tfrac{1}{p^*_{st}}, \mathrm{as} p^*_{st} = \prod_{i=0}^{n^*_j} w(v^*_i, v^*_{i+1}) \\ &< \log_2 \tfrac{1}{p_{st}}, \mathrm{as} 0 < p_{st} < p^*_{st} \leqslant 1 \\ &= d_{st}, \mathrm{as} p_{st} = \tfrac{1}{2^{d_{st}}} \end{aligned} \tag{A.1}$$

Therefore, P^*_{st} is a shorter path in $G\prime$ than P_{st} from s to t. This however contradicts the assumption that P_{st} is the shortest path. Then, the assumption that P_{st} is not the path in G from s to t with the maximum diffusion probability is incorrect. That is, P_{st} is the path with the maximum diffusion probability. The lemma then follows. □

References

[1] M. Amoruso, D. Anello, V. Auletta, R. Cerulli, D. Ferraioli, A. Raiconi, Contrasting the spread of misinformation in online social networks, J. Artif. Intell. Res. (JAIR) 69 (2020) 847–879.

[2] C. Arthur, What is the 1% rule? The guardian, Guardian News and Media (2006).

[3] V. Batagelj, M. Zaversnik, An O(m) algorithm for cores decomposition of networks, CoRR, vol. cs.DS/0310049, 2003. .

[4] C. Budak, D. Agrawal, A.E. Abbadi, Limiting the spread of misinformation in social networks, in: Proc. ACM Int. Conf. World Wide Web (WWW), 2015, pp. 665–674. .

[5] R.S. Burt, Structural holes: the social structure of competition, Harvard University Press, 1992.

[6] R.S. Burt, Structural holes and good ideas, Am. J. Sociol. 110 (2) (2004) 349–399.

[7] R.S. Burt, M. Kilduff, S. Tasselli, Social network analysis: foundations and frontiers on advantage, Annu. Rev. Psychol. 64 (2013) 527–547.

[8] R.S. Burt, Structural holes capstone, cautions, and enthusiasms, Classic Readings and New Directions in Egocentric Analysis, Cambridge University Press, Personal Networks, 2021.

[9] J. Cadena, D. Falcone, A. Marathe, A. Vullikanti, Discovery of under immunized spatial clusters using network scan statistics, BMC Medical Inf. Decision Making 19 (1) (2019), pp. 28:1–28:14.

[10] J. Cadena, A. Marathe, and A. Vullikanti, Finding spatial clusters susceptible to epidemic outbreaks due to undervaccination, in: Proc. 19th Int. Conf. Autonomous Agents MultiAgent Systems(AAMAS), 2020, pp. 1786–1788. .

[11] L. Chang, C. Zhang, X. Lin, and L. Qin, Scalable top-k structural diversity search, in: Proc. Int. Conf. Data Eng., 2017, pp. 95–98. .

[12] J.S. Coleman, Foundations of social theory, Harvard University Press, 1990.

[13] L.C. Freeman, A set of measures of centrality based on betweenness, Sociometry 40 (1) (1977) 35–41.

[14] M.L. Fredman, R.E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, ACM 34 (3) (1987) 596–615.

[15] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.

[16] Q. Gong, J. Zhang, X. Wang, and Y. Chen, Identifying structural hole spanners in online social networks using machine learning, in: Proc. ACM SIGCOMM Conf. Posters Demos., 2019, pp. 93–95. .

[17] S. Goyal, F. Vega-Redondo, Structural holes in social networks, Econ. Theor. 137 (1) (2007) 460–492.

[18] M.S. Granovetter, The strength of weak ties, Am. J. Sociol. 78 (6) (1973) 1360–1380.

[19] F. Guo, Y. Yuan, G. Wang, X. Zhao, and H. Su, Multi-attributed community search in road-social networks, in: Proc. 37th IEEE Int. Conf. Data Eng. (ICDE), 2021, pp. 109–120. .

[20] L. He, C. Lu, J. Ma, J. Cao, L. Shen, and P.S. Yu, Joint community and structural hole spanner detection via harmonic modularity, in: Proc. 22nd ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD), 2016, pp. 875–884. .

[21] D.S. Hochbaum, Approximation algorithms for NP-hard problems, ACM Sigact News 28 (2) (1997) 40–52.

[22] M. Kimura and K. Saito, Tractable models for information diffusion in social networks, in: Proc. 10th EUR Conf. Princ. Pract. Knowl. Discov. DBs (PKDD), 2006, pp. 259–271. .

[23] J. Leskovec and A. Krevl, SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data, 2014. .

[24] B. Liu, F. Zhang, W. Zhang, X. Lin, Y. Zhang, Efficient community search with size constraint, 37th IEEE Int. Conf. Data Eng. (ICDE) (2021) 97–108.

[25] T. Lou and J. Tang, Mining structural hole spanners through information diffusion in social networks, in: Proc. ACM Int. Conf. World Wide Web (WWW), 2013, pp. 825–836. .

[26] M.V. Marathe, A.K.S. Vullikanti, Computational epidemiology, Commun. ACM 56 (7) (2013) 88–96.

[27] T.V. Mierlo, The 1% rule in four digital health social networks: an observational study, J. Med. Internet Res. (JMIR) 16 (2) (2014) e33.

[28] F. Morone, H.A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (7563) (2015) 65–68.

[29] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions–I, Math. Programming 14 (1) (1978) 265–294.

[30] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.

[31] P.C. Pinto, P. Thiran, M. Vetterli, Locating the source of diffusion in large-scale networks, Phys. Rev. Lett. 109 (6) (2012) 068702.

[32] M. Rezvani, W. Liang, W. Xu, and C. Liu, Identifying top-k structural hole spanners in large-scale social networks, in: Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2015, pp. 263–272. .

[33] M.G. Rodriguez, J. Leskovec, and B. Schölkopf, Structure and dynamics of information pathways in online media, in: Proc. 6th ACM Int. Conf. Web Search Data Mining (WSDM), 2013, pp. 23–32. .

[34] K. Rost, The strength of strong ties in the creation of innovation, Res. Policy 40 (4) (2011) 588–604.

[35] N. Sun, P.-L.P. Rau, L. Ma, Understanding lurkers in online communities: a literature review, Comput. Hum. Behav. 38 (2014) 110–117.

[36] J. Tang, T. Lou, and J. Kleinberg, Inferring social ties across heterogeneous networks, in: Proc. 5th ACM Int. Conf. Web Search Data Mining (WSDM), 2012, pp. 743–752. .

[37] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (6380) (2018) 1146–1151.

[38] L. Wu, F. Morstatter, K.M. Carley, H. Liu, Misinformation in social media: definition, manipulation, and detection, ACM SIGKDD Explor. Newsl. 21 (2) (2019) 89–90.

[39] W. Xu, T. Li, W. Liang, J.X. Yu, N. Yang, S. Gao, Identifying structural hole spanners to maximally block information diffusion, Inf. Sci. 505 (2019) 100–126.

[40] W. Xu, M. Rezvani, W. Liang, J.X. Yu, C. Liu, Efficient algorithms for the identification of top-k structural hole spanners in large social networks, IEEE Trans. Knowl. Data Eng. (TKDE) 29 (5) (2017) 1017–1030.

[41] Y. Xu, H. Xu, D. Zhang, Y. Zhang, Finding overlapping community from social networks based on community forest model, Knowl. Based Syst. 109 (2016) 238–255.

[42] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowl. Inf. Syst. 42 (1) (2015) 181–213.

[43] Q.F. Ying, D.M. Chiu, and X. Zhang, Diversity of a user's friend circle in OSNs and its use for profiling, in: Proc. Int. Conf. Social Inf., 2018, pp. 471–486. .

[44] Y. Zhang, H. Xu, Y. Xu, J. Deng, J. Gu, R. Ma, J. Lai, J. Hu, X. Yu, L. Hou, L. Gu, Y. Wei, Y. Xiao, J. Lu, Finding structural hole spanners based on community forest model and diminishing marginal utility in large scale social networks, Knowl. Based Syst. 199 (2020) 105916.