



# Locating the propagation source in complex networks with observers-based similarity measures and direction-induced search

Fan Yang<sup>1,3</sup> · Chungui Li<sup>1</sup> · Yong Peng<sup>1</sup> · Jingxian Liu<sup>1,3</sup> · Yabing Yao<sup>2</sup> · Jiayan Wen<sup>1</sup> · Shuhong Yang<sup>1</sup>

Accepted: 24 February 2023 / Published online: 4 April 2023  
© The Author(s) 2023

## Abstract

Locating the propagation source is one of the most important strategies to control the harmful diffusion process on complex networks. Most existing methods only consider the infection time information of the observers, but the diffusion direction information of the observers is ignored, which is helpful to locate the source. In this paper, we consider both of the diffusion direction information and the infection time information to locate the source. We introduce a relaxed direction-induced search (DIS) to utilize the diffusion direction information of the observers to approximate the actual diffusion tree on a network. Based on the relaxed DIS, we further utilize the infection time information of the observers to define two kinds of observers-based similarity measures, including the Infection Time Similarity and the Infection Time Order Similarity. With the two kinds of similarity measures and the relaxed DIS, a novel source locating method is proposed. We validate the performance of the proposed method on a series of synthetic and real networks. The experimental results show that the proposed method is feasible and effective in accurately locating the propagation source.

**Keywords** Complex networks · Propagation source locating · Direction-induced search (DIS) · Observers-based similarity measures · Diffusion direction information · Infection time information

## 1 Introduction

In the modern world, the ubiquity of the diffusion phenomena taking on networks has incurred huge losses to human society. Some typical examples include computer virus propagation (Wang et al. 2014a), disease spreading (Zhang et al. 2018) and rumor diffusion (Hosseini and Azgomi 2016), etc. It is of great theoretical and practical significance to develop effective strategies to control the harmful diffusion process (Yu et al. 2017). As one of the significant measures, propagation source locating has attracted widespread

attentions, many effective methods are proposed in recent years (Jiang et al. 2017; Paluch et al. 2020). These methods can provide effective solutions for many important issues in reality, including locating the source(s) of SARS (Brockmann and Helbing 2013), COVID-19 (Tian et al. 2020), Cholera (Li et al. 2021), identifying the source of delay in public transportation networks (Manitz et al. 2017), estimating the source of foodborne disease (Horn and Friedrich 2019), etc.

It is well known that, when a diffusion process occurs on a network, there exists a spanning tree corresponding to the first time each node gets infected (Shah and Zaman 2011; Pinto et al. 2012; Tang et al. 2018). In fact, reconstructing the spanning tree is helpful to locate the propagation source (Yang et al. 2020). However, the commonly used breadth-first search (BFS) heuristic (Shah and Zaman 2011; Pinto et al. 2012; Yang et al. 2016) may be not an effective strategy (Tang et al. 2018; Yang et al. 2020). In this paper, we introduce an effective graph traversal method termed as relaxed direction-induced search (DIS), which is developed in our previous work (Yang et al. 2020). By utilizing the diffusion direction information of the observers, the relaxed DIS could effectively approximate the spanning tree correspond-

✉ Shuhong Yang  
yshgxust@163.com

Fan Yang  
100002022@gxust.edu.cn

<sup>1</sup> School of Computer Science and Technology, Guangxi University of Science and Technology, Liuzhou 545006, China

<sup>2</sup> School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

<sup>3</sup> Key Laboratory of Intelligent Information Processing and Graph Processing, Guangxi University of Science and Technology, Liuzhou 545006, China

ing to the first time each node gets infected. Based on the relaxed DIS, we further utilize the infection time information of the observers to define two kinds of observers-based similarity measures: (1) Infection Time Similarity, which measures the similarity between the observation infection time of the given observers and the measuring infection time of the given observers. (2) Infection Time Order Similarity, which measures the similarity between two sorted sequences of the given observers. One sequence is the observers ascending order obtained by sorting the observation infection time of the observers. Another sequence is the observers ascending order obtained by sorting the measuring infection time of the observers. Further, with the two kinds of similarity measures and the relaxed DIS, we propose a novel source locating method. Obviously, in the proposed method, both of the diffusion direction information and the infection time information are considered. Experiments are performed on a series of synthetic and real networks; the results show that the proposed method is feasible and effective in accurately locating the propagation source.

The rest of this paper is organized as follows. Existing related works are briefly reviewed in Sect. 2. We introduce the direction-induced search (DIS) in Sect. 3. Our method is proposed in Sect. 4. The performance of the proposed method is validated in Sect. 5. We conclude this work in Sect. 6.

## 2 Related work

For unweighted networks, a systematic method for propagation source locating was pioneered by Shah et al. (2011); they constructed a source estimator based on a novel topological quantity which is termed as Rumor Centrality (RC). Some researchers extended the RC to more complex environments, such as utilizing multiple observations to locate the source (Wang et al. 2014b), locating multi-sources (Luo et al. 2013; Wang et al. 2015) and so on. Zhu et al. (2016)

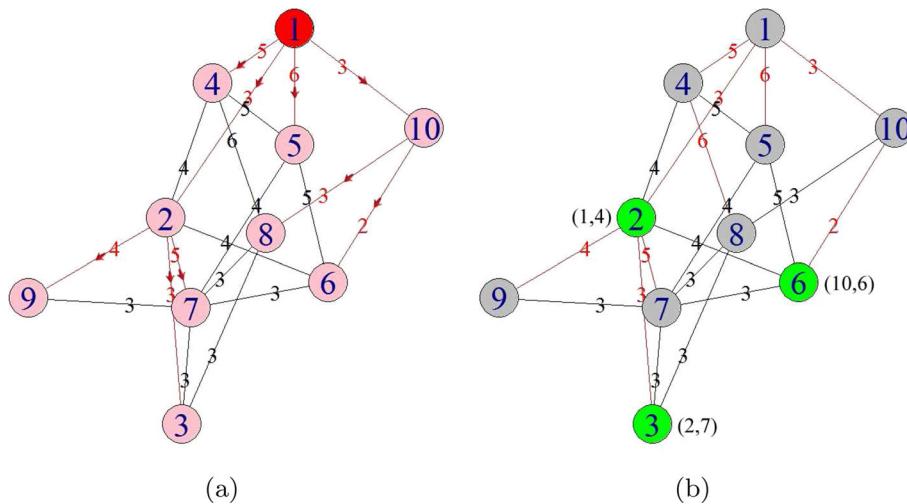
**Table 1** Notation summarization

Notation	Definition
$\mathcal{G}$	The topological graph of network
$\mathcal{V}$	The nodes set in $\mathcal{G}$
$\mathcal{E}$	The edges set in $\mathcal{G}$
$\theta$	The propagation delay set associated with $\mathcal{E}$
$\mathcal{O}$	Observers set
$o$	Observer
$K$	The number of observers
$s^*$	The propagation source
$\beta$	The propagation ratio of SI model
$ \cdot $	Calculating the number of element

developed a sample path-based method termed as Jordan Center (JC). Several improved methods based on the JC were developed to locate the source(s) with sparse observations (Zhu and Ying 2014; W.Luo et al. 2014; Jiang et al. 2018). Meanwhile, many source locating methods based on various ideas were developed for unweighted networks, including the Dynamic Message Passing-based method (Lokhov et al. 2014), the Belief Propagation base method (Altarelli et al. 2014), the Minimum Description Length-based method (Prakash et al. 2014), the Monte Carlo-based method (Antulov-Fantulin et al. 2015), the Rationality Observation-based method (Yang et al. 2016), the Time Aggregated Graph-based method (Chai et al. 2021), etc. The above methods are effective in unweighted networks. However, in reality, we have to consider various significant weights associated with the edges in networks, such as the traffic, the propagation delay and so on.

For weighted networks, Brockmann et al. (2013) modeled the Global Mobility Network as a weighted graph and proposed a source locating method based on a novel effective distance. This method is extended to more complex environments, including identifying the multiple sources (Jiang et al. 2015), identifying the source of delay in public transportation networks (Manitz et al. 2017), etc. However, the effective distance-based methods require the complete knowledge of nodes state. Meanwhile, there are several source locating methods based on various ideas for weighted networks (Cai et al. 2018; Chang et al. 2020; Feizi et al. 2019). But these methods also require the complete knowledge of nodes state. In reality, it is often the case that only limited nodes state can be observed (Caputo et al. 2019). To this problem, many methods were developed to locate the source with limited observers. Shen et al. (2016) developed a time-reversal backward spreading (TRBS) algorithm, but this algorithm may not work if the locatability condition is violated. Hu et al. (2019) proposed a greedy optimization algorithm to reduce the number of observers for TRBS. Tang and Ji et al. (2018) et al. proposed a source estimation algorithm based on the Gromov matrix. However, the Gromov matrix may be not the optimal heuristic for source locating. Meanwhile, Fu et al. (2016) proposed a backward diffusion-based method for multiple sources locating. Wang (2019) and Xu et al. (2019) identified the diffusion source based on the Spearman's coefficient. Wang and Sun (2020) proposed a sequential neighbor filtering (SNF) algorithm for heterogeneous propagation models. Wang et al. (2021) proposed three source locating algorithms by defining the estimated mean and standard deviation of the propagation delay. However, the methods using limited observers only considered the infection time information, but the diffusion direction information was ignored.

The Gaussian estimator (Pinto et al. 2012) first located the source with limited observers by utilizing the diffusion direction information of the observers, and its time complexity can



**Fig. 1** Sub-Fig. 1a shows the SI model ( $\beta = 1$ ) diffused on a given network. The infection is initiated by node 1 (with “red” color). All the “red” paths form a diffusion tree (rooted at node 1) of the network. The arrows attached to the “red” paths represent the actual diffusion direction. The infection diffused along this actual diffusion tree. The nodes with “pink” color are in the infectious state. Sub-Fig. 1b shows a relaxed DIS spanning tree (rooted at node 1) of the network in Sub-Fig. 1a. This

**Table 2** The time complexity of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms

Algorithm	Time complexity
Gauss	$O( \mathcal{V} ^3)$ (Pinto et al. 2012)
GSSI	$O( \mathcal{V} ^3 +  \mathcal{V} \mathcal{K}^3)$ (Tang et al. 2018)
TRBS	$O(\mathcal{K}\mathcal{N} \log \mathcal{N})$ (Shen et al. 2016)
SNF	$O( \mathcal{V} ^2 D )$ (Wang et al. 2022)
OSBFS	$O( \mathcal{V} ^3 + z \mathcal{V} ^2\mathcal{K})$
OSDIS	$O( \mathcal{V} ^3 + z \mathcal{V} ^2\mathcal{K})$

**Table 3** The parameters for generating the synthetic networks

Networks	Parameters
BA model (1)	barabasi.game (200, power=1.5, m=2, directed=False)
BA model (2)	barabasi.game (200, power=1.7, m=2, directed=False)
BA model (3)	barabasi.game (200, power=1.9, m=2, directed=False)
BA model (4)	barabasi.game (200, power=2.1, m=2, directed=False)
BA model (5)	barabasi.game (200, power=2.3, m=2, directed=False)
BA model (6)	barabasi.game (200, power=2.5, m=2, directed=False)
WS model (1)	watts.strogatz.game (1, 200, 2, 0.2)
WS model (2)	watts.strogatz.game (1, 200, 2, 0.4)
WS model (3)	watts.strogatz.game (1, 200, 2, 0.6)
WS model (4)	watts.strogatz.game (1, 200, 2, 0.8)
WS model (5)	watts.strogatz.game (1, 200, 2, 1.0)

---

Software environment: R 64 × 3.3.3 + igraph R 1.2.1

be reduced by ignoring the observers with low-quality information (Paluch et al. 2018). However, the diffusion direction

tree is constructed by the edges with “red” color, which is generated by the relaxed DIS algorithm with three observers (nodes 2, 3 and 6, with “green” color). The nodes with “gray” color cannot be observed. The pair of value next to each observer represents the recorded Diffusion Direction information and Diffusion Timing information (color figure online)

information is only used in the tree graphs. In our previous work (Yang et al. 2020), a relaxed direction-induced search (DIS) was proposed by utilizing the diffusion direction information. With the relaxed DIS, the accuracy of the Gaussian estimator on general graphs is improved. Different from the previous work, in this paper, we first introduce the relaxed direction-induced search (DIS) (Yang et al. 2020) to utilize the diffusion direction information of the observers to approximate the actual diffusion tree on a network. Based on the relaxed DIS, we further utilize the infection time information of the observers to define two kinds of similarity measures, including the Infection Time Similarity and the Infection Time Order Similarity. With the two kinds of similarity measures and the relaxed DIS, we propose a novel source locating method. Obviously, the diffusion direction information and the infection time information are combined in this method. The feasibility and effectiveness of this method are validated on a series of synthetic and real networks.

3 Preliminaries

A network is modeled as an undirected and weighted graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{\theta})$ , where  $\mathcal{V}$  and  $\mathcal{E}$  represent the nodes set and edges set, respectively.  $\boldsymbol{\theta} = \{\theta_{uv}\}$ , where  $\theta_{uv}$  denotes the random propagation delay associated with an edge connecting nodes  $u$  and  $v$ ,  $u, v \in \mathcal{V}$ ,  $vu \in \mathcal{E}$ . The random variables  $\theta_{vu}$  for different edges  $vu$  have a known, arbitrary joint distribution.

*Diffusion model* Similar to the references (Zhu and Ying 2016; W.Luo et al. 2014; Lokhov et al. 2014; Yang et al.

2016), the diffusion process on  $\mathcal{G}$  is discrete. We adopt a simple Susceptible-Infectious (SI) model. With the SI model, each node in  $\mathcal{V}$  is only in one of the two states: (1) susceptible, if it has not been infected so far, or (2) infectious, if it has been infected by any one neighbor. The diffusion process on  $\mathcal{G}$  is initiated by a single propagation source (denoted by  $s^*$ ) at an unknown time  $t^*$ . All nodes are susceptible except for  $s^*$  is infectious. A diffusion is possible from an infected node to a susceptible node if and only if there is an edge between them. Once infected, the node will stay the infectious state forever. Let  $\mathcal{N}(v)$  denote the neighbors set of node  $v$ , suppose  $v$  is infected by one neighbor  $w$  at time  $t_v$ , then  $v$  will attempt to infect each susceptible neighbor  $u \in \mathcal{N}(v)$  (except for  $w$ ) along the weighted edge  $vu$  with propagation

**Table 4** Real networks

Networks	Data source
Dolphins	<a href="http://networkrepository.com/dolphins.php">http://networkrepository.com/dolphins.php</a>
Lesmis	<a href="http://networkrepository.com/lesmis.php">http://networkrepository.com/lesmis.php</a>
PDZBase(LCC)	<a href="http://konect.cc/networks/maayan-pdzbase/">http://konect.cc/networks/maayan-pdzbase/</a>
USAirlines	<a href="http://networkrepository.com/USAir97.php">http://networkrepository.com/USAir97.php</a>
NetScience(LCC)	<a href="http://konect.cc/networks/dimacs10-netscience/">http://konect.cc/networks/dimacs10-netscience/</a>
Celegans	<a href="http://konect.cc/networks/dimacs10-celegans_metabolic/">http://konect.cc/networks/dimacs10-celegans_metabolic/</a>
Euroroads(LCC)	<a href="http://konect.cc/networks/subelj_euroroad/">http://konect.cc/networks/subelj_euroroad/</a>

LCC the abbreviation of largest connected component

**Table 5** The topological properties of the used networks

Network	$ \mathcal{V} $	$ \mathcal{E} $	$\langle k \rangle$	A	APL
BA model (1)	200	397	3.97	-0.321	2.85
BA model (2)	200	397	3.97	-0.532	2.43
BA model (3)	200	397	3.97	-0.501	2.13
BA model (4)	200	397	3.97	-0.807	2.04
BA model (5)	200	397	3.97	-0.757	2.02
BA model (6)	200	397	3.97	-0.806	2.00
WS model (1)	200	400	4.00	0.102	4.39
WS model (2)	200	400	4.00	0.0003	4.04
WS model (3)	200	400	4.00	0.044	4.03
WS model (4)	200	400	4.00	-0.059	3.96
WS model (5)	200	400	4.00	-0.051	3.91
Dolphins	62	159	5.13	-0.044	3.36
Lesmis	77	254	6.60	-0.165	2.64
PDZBase	161	209	2.60	-0.466	5.33
USAirlines	332	2126	12.81	-0.208	2.74
Netscience	379	914	4.82	-0.082	6.04
Celegans	453	2025	8.94	-0.226	2.66
Euroroads	1039	1305	2.51	0.090	18.40

$\langle k \rangle$  the average degree of  $\mathcal{G}$

A the assortative coefficient (Newman 2002)

APL the average path length (the number of edges)

ratio  $\beta$ . If there are two or more infected neighbors having a same propagation delay to  $u$ ,  $u$  can be first time infected by only one neighbor. Without loss of generality, the diffusion process is terminated when there are no susceptible nodes in  $\mathcal{G}$ .

Let  $\mathcal{O} = \{o_k\}_{k=1}^K \subseteq \mathcal{V}$  denote the set of  $K$  observable nodes on  $\mathcal{G}$ , termed as observers set, whose location in  $\mathcal{G}$  is known. Generally, there is  $K \ll |\mathcal{V}|$ . Similar to the references (Pinto et al. 2012; Yang et al. 2020), each  $o_k \in \mathcal{O}$  can provide two types of information: (1) the Diffusion Direction information in which the infection arrives to  $o_k$ , (2) the Infection Timing information at which the infection arrives to  $o_k$ .

**Algorithm 1** The relaxed direction-induced search (DIS) algorithm

```

Input: an arbitrary  $\mathcal{G}$  and  $\mathcal{O}$ 
Output:  $T_{\text{DIS},s}$ 
1: for each node  $s \in \mathcal{V}$  do
2:   initialize an empty set  $E(\mathcal{T})$ .
3:   initialize an empty queue  $\mathcal{Q}$ .
4:    $\mathcal{Q}.\text{enqueue}(s)$  and set  $s$  as visited.
5:   while  $\mathcal{Q}$  is not empty do
6:      $v = \mathcal{Q}.\text{front}()$ .
7:      $\mathcal{Q}.\text{dequeue}()$ .
8:     for each neighbor  $u$  of  $v$  do
9:       if  $u$  has not been visited then
10:        if  $u \in \mathcal{O}$  then
11:          if the Diffusion Direction information recorded in  $u$  is  $v$  then
12:             $\mathcal{Q}.\text{enqueue}(u)$ , set  $u$  as visited,  $E(\mathcal{T}) = E(\mathcal{T}) \cup e_{v \rightarrow u}$ .
13:          end if
14:        else
15:           $\mathcal{Q}.\text{enqueue}(u)$ , set  $u$  as visited,  $E(\mathcal{T}) = E(\mathcal{T}) \cup e_{vu}$ .
16:        end if
17:      end if
18:    end for
19:  end while
20:  remove the edges  $\notin E(\mathcal{T})$  from  $\mathcal{G}$  and mark the remaining part of  $\mathcal{G}$  as  $\mathcal{T}$ .
21:  if  $|E(\mathcal{T})| == |\mathcal{V}| - 1$  then
22:    mark  $\mathcal{T}$  as DIS spanning tree  $T_{\text{DIS},s}$  rooted at node  $s$ .
23:  end if
24: end for

```

The  $e_{v \rightarrow u}$  in line 12 represents the actual diffusion direction between  $v$  and  $u$ , which indicates that the infection is the first time diffused from  $v$  to  $u$ . The  $e_{vu}$  in line 15 represents that the diffusion direction is assumed from  $v$  to  $u$ . The  $T_{\text{DIS},s}$  in line 22 denotes a relaxed DIS spanning tree rooted at node  $s$ .

**Direction-induced search (DIS)** We introduce a graph traversal method termed as relaxed direction-induced search (DIS), which is developed in our previous work (Yang et al. 2020). The relaxed DIS is summarized in Algorithm 1. The  $E(\mathcal{T})$  declared in line 2 is an edge set. The function of lines 3–19 is to traverse the  $\mathcal{G}$  with  $s$  as root, which requires  $O(|\mathcal{V}|^2)$  computations in the worst case. Here, from lines 10–16, we know that, if a node is an observer, then the infection direction is determined by the Diffusion Direction information recorded in this node. If the node is a non-observer, the infection direction will be assumed to be its current parent node. From lines 20–23, a DIS spanning tree  $T_{\text{dis},s}$  is generated if and only if “ $|E(\mathcal{T})| == |\mathcal{V}| - 1$ ”, where line 20 requires  $O(|\mathcal{V}| + |\mathcal{E}|)$  computations. Finally, taking the loop in line 1 into account, the time complexity of Algorithm 1 is  $O(|\mathcal{V}|^3)$ . Further, by

using Algorithm 1, a relaxed DIS spanning tree is generated by utilizing the Diffusion Direction information recorded in  $\mathcal{O}$ .

Frequently used notations are summarized in Table 1.

## 4 The proposed method

Given an arbitrary  $\mathcal{G}$  and an arbitrary  $\mathcal{O}$ , we locate the propagation source by measuring the similarity between the  $\mathcal{O}$  in the actual diffusion tree (corresponding to the first time each node gets infected) and the  $\mathcal{O}$  in a spanning tree of  $\mathcal{G}$ , which can be described as an estimator that maximizes the similarity.

$$\hat{s} = \arg \max_{s \in \mathcal{V}} \mathcal{S}(\mathcal{O}_{\mathcal{T}_{s^*}}, \mathcal{O}_{\mathcal{T}_s}) \quad (1)$$

where  $\mathcal{T}_{s^*}$  denotes the actual diffusion tree with source  $s^*$  as root, and  $\mathcal{T}_s$  denotes a tree that spans all nodes in  $\mathcal{G}$  with node  $s$  as root.  $\mathcal{O}_{\mathcal{T}_{s^*}}$  and  $\mathcal{O}_{\mathcal{T}_s}$  denote the given  $\mathcal{O}$  in  $\mathcal{T}_{s^*}$  and  $\mathcal{T}_s$ , respectively.  $\mathcal{S}(\mathcal{O}_{\mathcal{T}_{s^*}}, \mathcal{O}_{\mathcal{T}_s})$  denotes the similarity between  $\mathcal{O}_{\mathcal{T}_{s^*}}$  and  $\mathcal{O}_{\mathcal{T}_s}$ .

Theoretically, we have to evaluate  $\mathcal{S}(\mathcal{O}_{\mathcal{T}_{s^*}}, \mathcal{O}_{\mathcal{T}_s})$  in Eq. 1 for all spanning trees of  $\mathcal{G}$  and then select the one with

the maximal similarity and its root is the  $s^*$ . However, the complexity to generate all spanning trees of  $\mathcal{G}$  will increase exponentially with the number of nodes. Therefore, we introduce an approximation by assuming that the actual diffusion tree is a relaxed DIS spanning tree (obtained by Algorithm 1), and the time complexity only requires  $O(|\mathcal{V}|^3)$ . Then, Eq. 1 can be modified as follows.

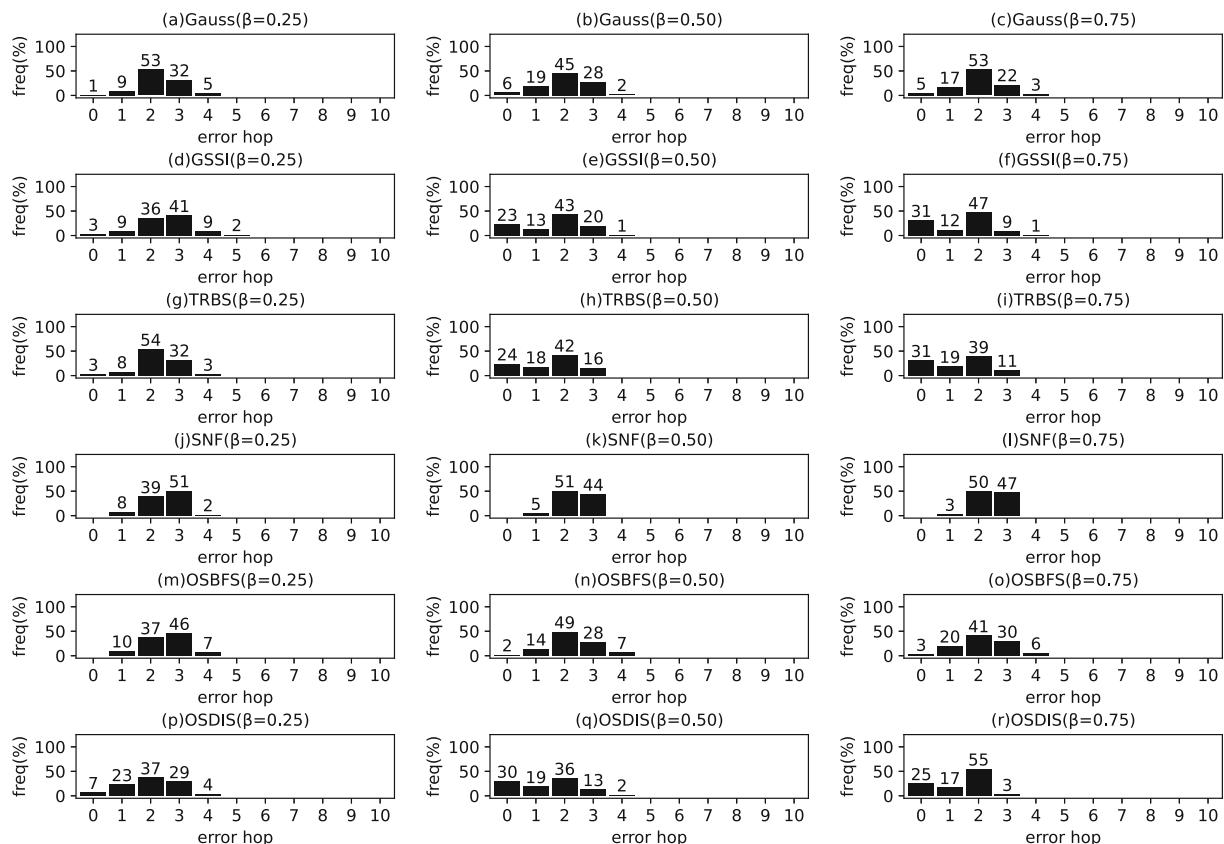
$$\hat{s} = \arg \max_{s \in \mathcal{V}} \mathcal{S}(\mathcal{O}_{\mathcal{T}_{s^*}}, \mathcal{O}_{\mathcal{T}_{\text{DIS},s}}) \quad (2)$$

where  $\mathcal{T}_{\text{DIS},s}$  is a relaxed DIS spanning tree of  $\mathcal{G}$  with a node  $s$  as root.  $\mathcal{O}_{\mathcal{T}_{\text{DIS},s}}$  denote the given  $\mathcal{O}$  in  $\mathcal{T}_{\text{DIS},s}$ .  $\mathcal{S}(\mathcal{O}_{\mathcal{T}_{s^*}}, \mathcal{O}_{\mathcal{T}_{\text{DIS},s}})$  denotes the similarity between  $\mathcal{O}_{\mathcal{T}_{s^*}}$  and  $\mathcal{O}_{\mathcal{T}_{\text{DIS},s}}$ .

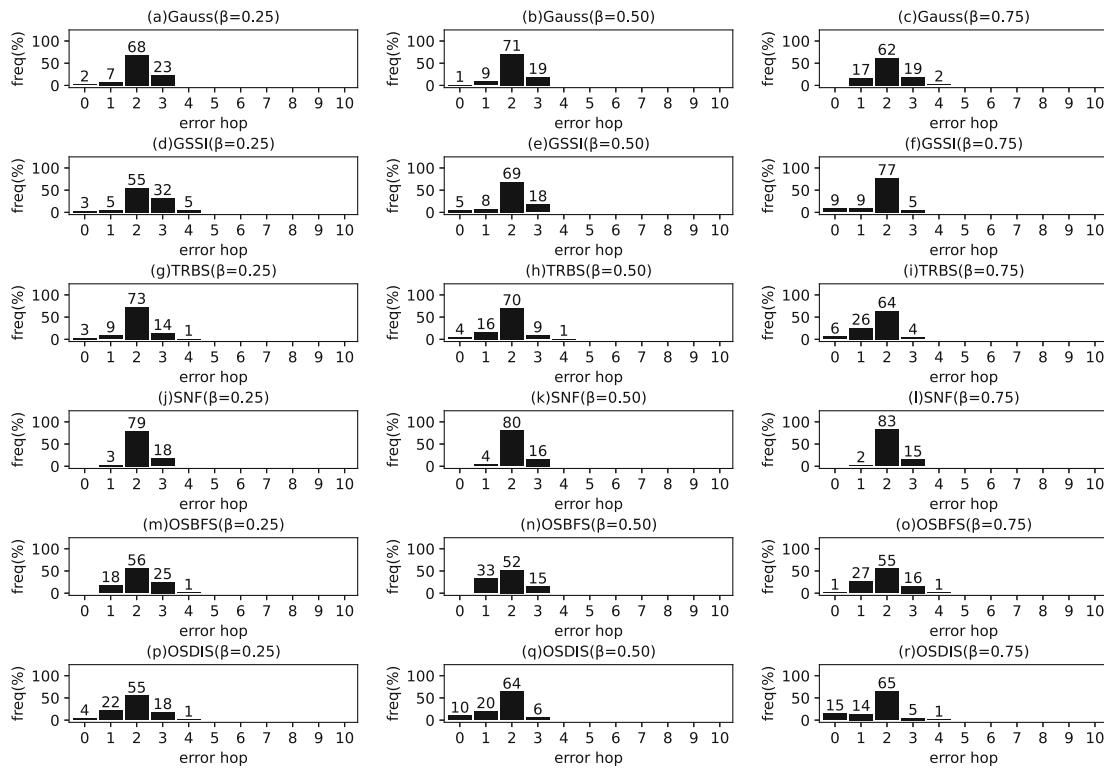
Since  $\mathcal{K} < |\mathcal{V}|$ ,  $\mathcal{T}_{\text{DIS},s}$  may be not unique, and each  $\mathcal{T}_{\text{DIS},s}$  may not correspond to the actual diffusion tree. Thus, the relaxed DIS is obviously a sub-optimal heuristic.

### 4.1 Observers-based similarity measures

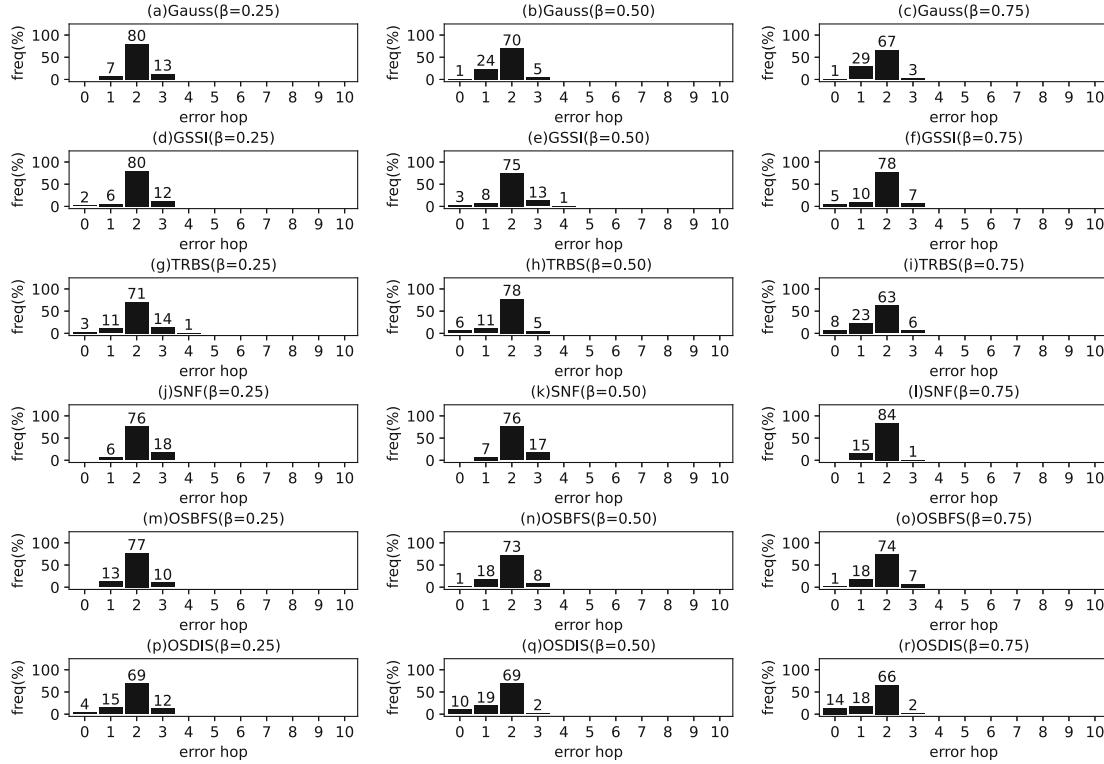
In this subsection, we first define two kinds of observers-based similarity measures by utilizing the infection time information of observers. One is Infection Time Similarity; another is Infection Time Order Similarity.



**Fig. 2** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on BA model (1). Each sub-figure is obtained by 100 runs



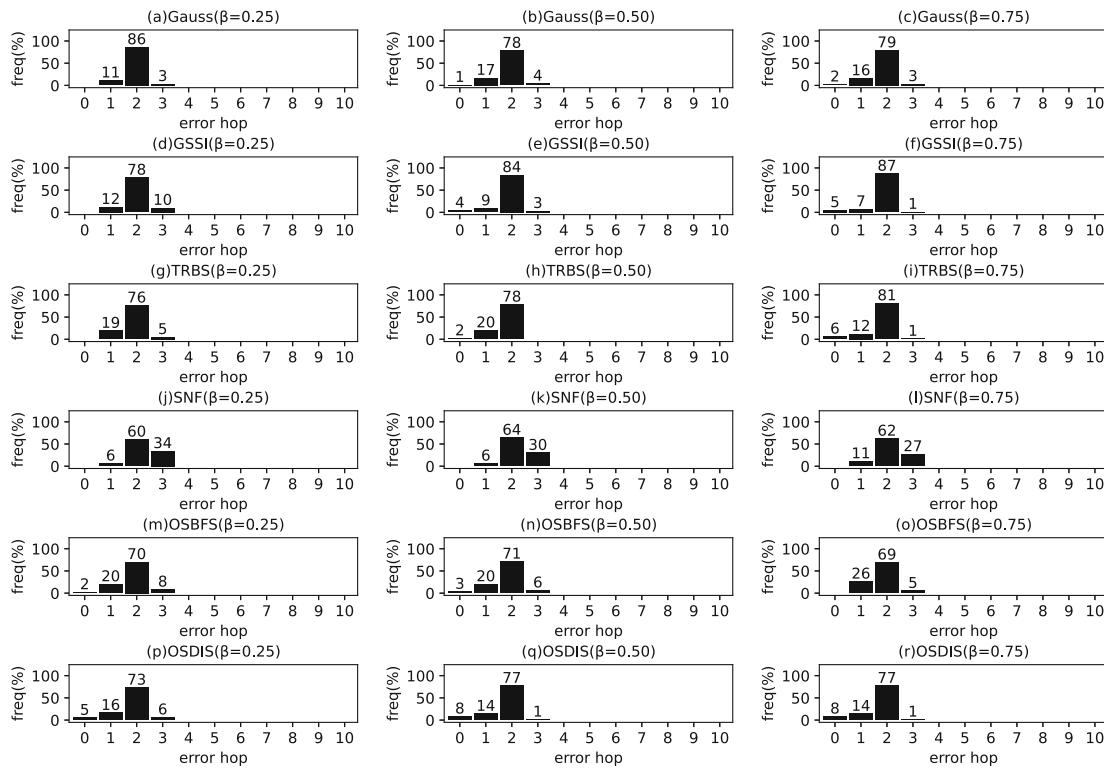
**Fig. 3** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on BA model (2). Each sub-figure is obtained by 100 runs



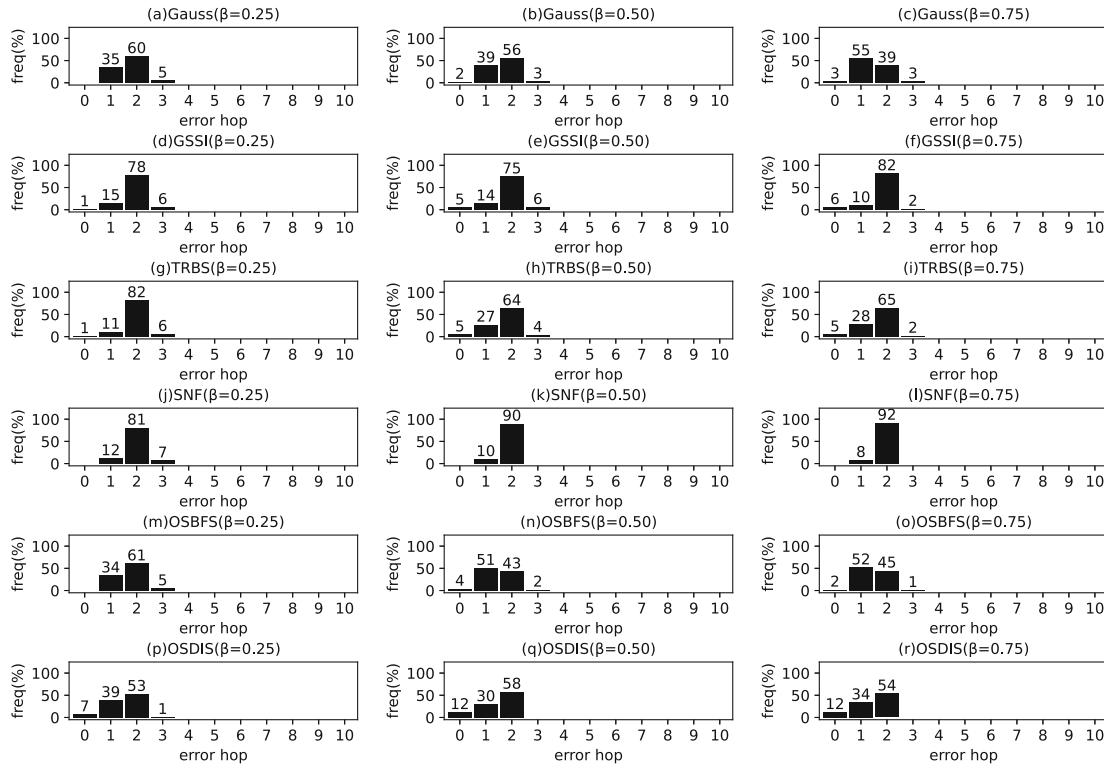
**Fig. 4** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on BA model (3). Each sub-figure is obtained by 100 runs

**Definition 1** Observation Infection Time. Given an arbitrary  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \theta)$  and an observers set  $\mathcal{O} = \{o_k\}_{k=1}^{\mathcal{K}}$ . the Obser-

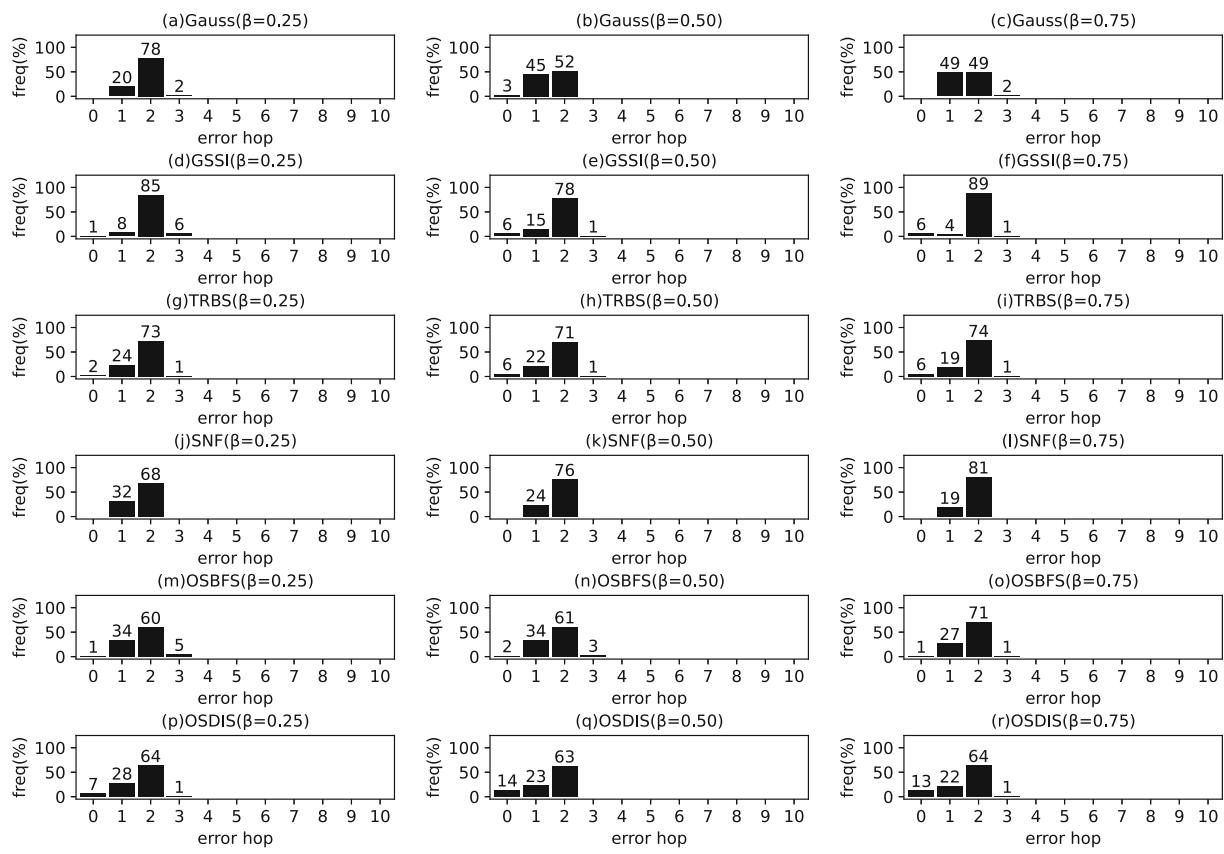
vation Infection Time of  $\mathcal{O}$  is defined as  $\mathbf{T}_{\mathcal{O}} = \{t_{o_k}\}_{k=1}^{\mathcal{K}}$ ,



**Fig. 5** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on BA model (4). Each sub-figure is obtained by 100 runs



**Fig. 6** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on BA model (5). Each sub-figure is obtained by 100 runs



**Fig. 7** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on BA model (6). Each sub-figure is obtained by 100 runs

**Table 6** The average error hop of the six algorithms on different networks

$\beta$	Network	Gauss	GSSI	TRBS	SNF	OSBFS	OSDIS
0.25	BA model (1)	2.31	2.50	2.24	2.47	2.50	<b>2.00</b>
	BA model (2)	2.12	2.31	2.01	2.15	2.09	<b>1.90</b>
	BA model (3)	2.06	2.02	1.99	2.12	1.97	<b>1.89</b>
	BA model (4)	1.92	1.98	1.86	2.28	1.84	<b>1.80</b>
	BA model (5)	1.70	1.89	1.93	1.95	1.71	<b>1.48</b>
	BA model (6)	1.82	1.96	1.73	1.68	1.69	<b>1.59</b>
	WS model (1)	2.61	2.34	2.34	2.93	2.77	<b>1.81</b>
	WS model (2)	2.60	2.26	2.10	3.08	2.84	<b>1.78</b>
	WS model (3)	2.69	2.26	2.18	2.99	3.62	<b>1.80</b>
	WS model (4)	2.61	2.30	2.14	2.89	3.39	<b>1.64</b>
	WS model (5)	2.32	2.12	1.86	2.79	3.56	<b>1.85</b>
	Dolphins	2.11	2.58	1.98	2.56	2.76	<b>1.60</b>
	Lesmis	2.19	2.16	2.08	2.60	2.10	<b>1.70</b>
	PDZBase	3.07	2.32	2.27	3.42	3.70	<b>1.99</b>
	USAirlines	2.26	2.19	2.19	2.29	2.73	<b>1.89</b>
0.50	NetScience	3.30	2.06	2.39	2.74	3.28	<b>1.94</b>
	Celegans	2.22	2.26	2.11	2.22	2.39	<b>1.84</b>
	Euroroads	6.28	<b>3.44</b>	6.08	6.71	12.01	4.47
	BA model(1)	2.01	1.63	1.50	2.39	2.24	<b>1.38</b>
0.75	BA model (2)	2.08	2.00	1.87	2.12	1.82	<b>1.66</b>
	BA model (3)	1.79	2.01	1.82	2.10	1.88	<b>1.63</b>

**Table 6** continued

$\beta$	Network	Gauss	GSSI	TRBS	SNF	OSBFS	OSDIS
0.75	BA model (4)	1.85	1.86	1.76	2.24	1.80	<b>1.71</b>
	BA model (5)	1.60	1.82	1.67	1.90	<b>1.43</b>	1.46
	BA model (6)	<b>1.49</b>	1.74	1.67	1.76	1.65	<b>1.49</b>
	WS model (1)	2.12	1.35	0.98	2.97	1.76	<b>0.88</b>
	WS model (2)	2.20	1.25	<b>1.01</b>	2.85	2.78	1.23
	WS model (3)	2.06	1.29	1.21	2.86	3.37	<b>1.17</b>
	WS model (4)	2.21	1.06	0.97	2.80	3.18	<b>0.95</b>
	WS model (5)	2.17	1.30	1.35	3.03	3.49	<b>1.10</b>
	Dolphins	2.21	2.19	1.68	2.74	2.47	<b>1.37</b>
	Lesmis	2.10	1.51	1.66	2.70	1.77	<b>1.39</b>
	PDZBase	2.57	1.65	1.71	3.38	3.04	<b>1.47</b>
	USAirlines	2.21	1.83	1.84	2.06	2.31	<b>1.80</b>
	NetScience	3.07	1.77	2.03	2.58	2.36	<b>1.55</b>
	Celegans	1.98	1.78	1.74	2.29	2.24	<b>1.56</b>
	Euroroads	6.52	2.56	4.22	6.69	8.14	<b>2.53</b>
	BA model(1)	2.01	1.37	1.50	2.44	2.16	<b>1.36</b>
	BA model (2)	2.06	1.78	1.66	2.13	1.89	<b>1.63</b>
	BA model (3)	1.72	1.87	1.67	1.86	1.87	<b>1.56</b>
	BA model (4)	1.83	1.84	1.77	2.16	1.79	<b>1.71</b>
	BA model (5)	<b>1.42</b>	1.80	1.64	1.92	1.45	<b>1.42</b>
	BA model (6)	<b>1.53</b>	1.85	1.70	1.81	1.72	<b>1.53</b>
	WS model (1)	1.82	<b>0.29</b>	0.36	2.96	1.90	0.66
	WS model (2)	1.90	0.37	<b>0.29</b>	2.68	2.28	0.79
	WS model (3)	2.22	<b>0.46</b>	0.51	2.77	3.15	1.03
	WS model (4)	1.86	0.48	<b>0.25</b>	2.69	3.04	1.00
	WS model (5)	1.80	0.48	<b>0.42</b>	2.78	3.25	0.85
	Dolphins	1.95	2.00	1.45	2.85	2.13	<b>1.29</b>
	Lesmis	2.00	1.56	1.27	2.75	1.61	<b>1.22</b>
	PDZBase	2.68	1.37	1.45	3.34	2.56	<b>1.31</b>
	USAirlines	2.10	<b>1.24</b>	1.36	1.98	2.21	1.52
	NetScience	3.02	1.44	1.54	2.38	1.92	<b>1.37</b>
	Celegans	1.98	1.08	<b>1.05</b>	1.92	2.27	1.29
	Euroroads	5.89	1.78	2.29	7.12	3.74	<b>1.64</b>

The values with bold font are the optimal values

where  $t_{o_k}$  denotes the Infection Timing information recorded in  $o_k$ .

**Definition 2** Measuring Infection Time. Given an arbitrary  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{\theta})$  and an observers set  $\mathcal{O} = \{o_k\}_{k=1}^K$ . The Measuring Infection Time of  $\mathcal{O}$  in  $\mathcal{T}_{\text{DIS},s}$  is defined as:

$$\mathbf{T}_{\mathcal{T}_{\text{DIS},s}} = \left\{ t_{v_{o_k}} \right\}_{k=1}^K \quad (3)$$

where  $\mathcal{T}_{\text{DIS},s}$  is a relaxed DIS spanning tree of  $\mathcal{G}$  with a node  $s$  as root,  $s \in \mathcal{V}$ .  $t_{v_{o_k}}$  denotes the Measuring Infection Time of node  $v_{o_k}$ ;  $v_{o_k}$  is the node with the node number corresponding

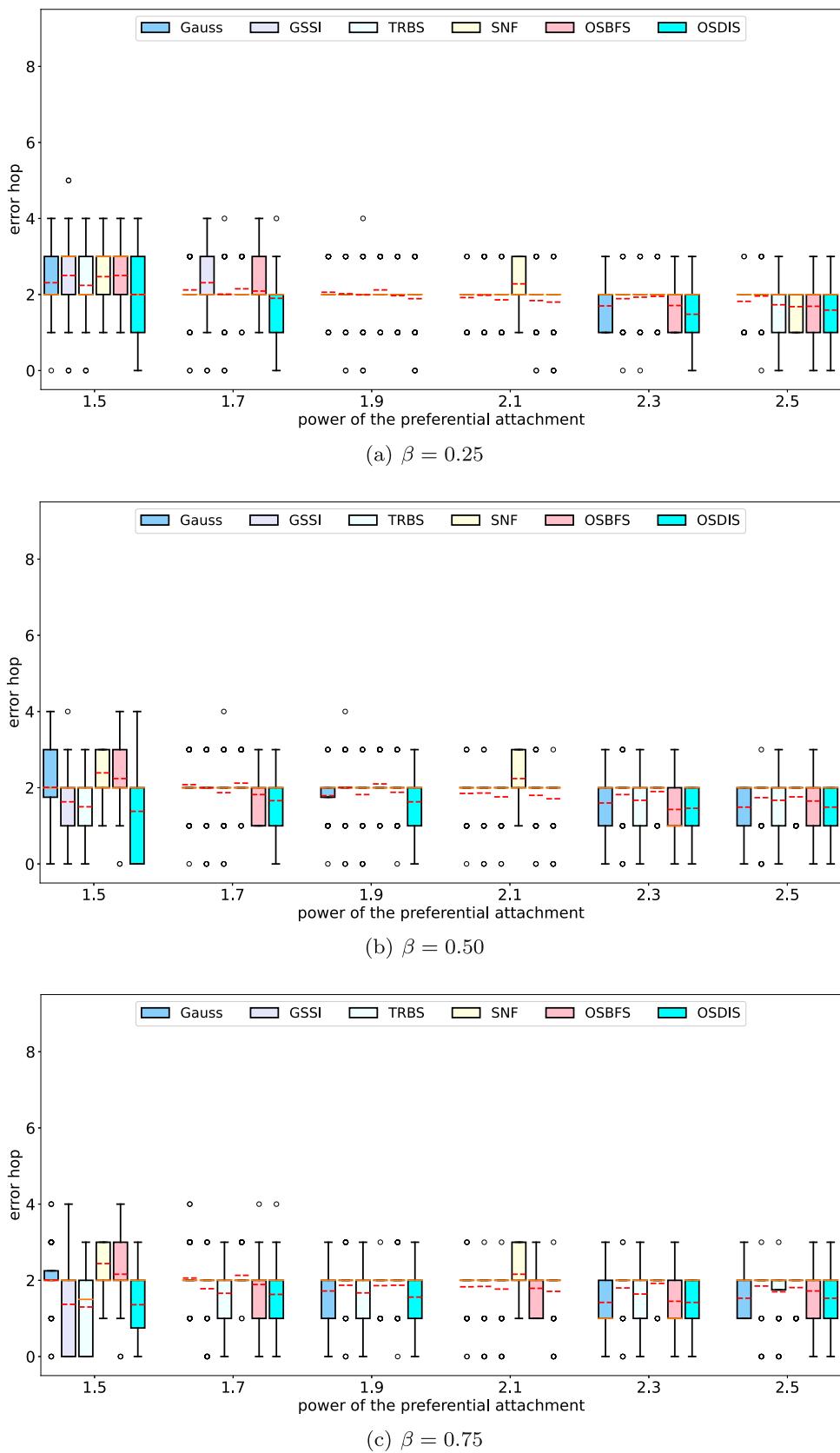
to the observer  $o_k$ .

$$t_{v_{o_k}} = \begin{cases} t^*, & v_{o_k} = s \\ t^* + \sum_{j \in p(s, v_{o_k})} \theta_j, & v_{o_k} \neq s \end{cases} \quad (4)$$

where  $t^*$  is the unknown start time.  $s$  is the root of  $\mathcal{T}_{\text{DIS},s}$ .  $p(s, v_{o_k})$  denotes the path from  $s$  to  $v_{o_k}$  in the  $\mathcal{T}_{\text{DIS},s}$ .  $\theta_j$  denotes the propagation delay.

In fact, in the current fixed  $\mathcal{T}_{\text{DIS},s}$  with  $s$  as root, the  $s$  is assumed to be the propagation source. Thus,  $t^*$  minimizes the difference between  $\mathbf{T}_{\mathcal{O}}$  and  $\mathbf{T}_{\mathcal{T}_{\text{DIS},s}}$ ;  $t^*$  can be estimated

**Fig. 8** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on the BA models generated by different powers of the preferential attachment. The orange line and red line in each box denote the median and average error hop (also shown in Table 6), respectively (color figure online)



**Table 7** The average error delay of the six algorithms on different networks

$\beta$	Network	Gauss	GSSI	TRBS	SNF	OSBFS	OSDIS
0.25	BA model (1)	8.56	9.07	8.29	8.94	9.56	<b>7.03</b>
	BA model (2)	7.89	8.33	7.52	8.13	7.92	<b>6.95</b>
	BA model (3)	7.68	8.01	7.59	8.39	7.63	<b>7.20</b>
	BA model (4)	6.95	7.64	6.87	9.21	7.00	<b>6.51</b>
	BA model (5)	5.98	7.11	7.14	7.53	6.54	<b>5.33</b>
	BA model (6)	6.66	7.54	6.36	6.64	6.56	<b>5.73</b>
	WS model (1)	9.61	8.80	8.57	11.34	10.40	<b>6.73</b>
	WS model (2)	10.04	8.54	7.79	11.87	10.99	<b>6.53</b>
	WS model (3)	10.09	8.57	8.11	11.37	13.23	<b>6.48</b>
	WS model (4)	9.74	8.72	8.11	11.71	13.02	<b>6.24</b>
	WS model (5)	8.75	7.89	7.04	10.92	13.49	<b>6.81</b>
	Dolphins	7.68	9.53	7.39	9.13	9.60	<b>5.79</b>
	Lesmis	8.22	7.82	7.53	9.68	7.86	<b>6.36</b>
	PDZBase	12.07	8.94	8.90	13.60	14.24	<b>7.54</b>
	USAirlines	8.01	7.82	7.70	8.09	9.95	<b>6.74</b>
	NetScience	13.00	7.98	9.31	11.33	12.82	<b>7.62</b>
	Celegans	7.70	7.77	7.29	7.89	8.36	<b>6.40</b>
	Euroroads	24.93	<b>13.59</b>	24.48	27.00	47.75	17.93
0.50	BA model(1)	7.60	6.01	5.60	9.07	8.49	<b>4.88</b>
	BA model (2)	7.51	7.64	7.03	8.05	6.87	<b>5.92</b>
	BA model (3)	6.71	7.43	6.93	8.31	7.10	<b>6.05</b>
	BA model (4)	7.20	6.90	6.47	9.04	6.92	<b>6.05</b>
	BA model (5)	5.75	6.49	5.93	7.13	5.32	<b>5.00</b>
	BA model (6)	5.41	6.28	5.78	6.98	6.24	<b>5.20</b>
	WS model (1)	8.02	4.89	3.57	11.83	6.27	<b>3.16</b>
	WS model (2)	8.17	4.69	<b>3.87</b>	10.89	10.48	4.52
	WS model (3)	7.21	4.64	4.38	10.70	12.61	<b>4.26</b>
	WS model (4)	8.32	4.05	3.82	11.39	12.29	<b>3.59</b>
	WS model (5)	8.06	4.97	5.02	11.63	12.94	<b>3.97</b>
	Dolphins	7.97	8.16	6.18	9.95	8.79	<b>4.92</b>
	Lesmis	7.73	5.48	6.14	10.57	6.70	<b>5.39</b>
	PDZBase	10.16	6.24	6.50	13.21	11.61	<b>5.54</b>
	USAirlines	7.55	<b>6.35</b>	6.52	7.33	8.39	6.65
	NetScience	12.02	6.71	7.73	10.46	9.22	<b>6.10</b>
	Celegans	6.87	5.96	5.93	7.67	7.86	<b>5.27</b>
	Euroroads	26.12	10.20	16.90	27.07	32.59	<b>9.97</b>
0.75	BA model(1)	7.62	4.89	<b>4.58</b>	9.34	8.04	4.84
	BA model (2)	7.48	6.76	6.24	8.14	7.05	<b>5.66</b>
	BA model (3)	6.35	6.95	6.28	7.49	7.08	<b>5.64</b>
	BA model (4)	7.18	6.77	6.54	8.63	6.74	<b>5.84</b>
	BA model (5)	5.10	6.38	5.86	7.58	5.29	<b>4.98</b>
	BA model (6)	5.67	6.56	5.98	7.26	6.47	<b>5.20</b>
	WS model (1)	6.34	<b>0.99</b>	1.17	11.27	6.79	2.19
	WS model (2)	7.04	1.22	<b>1.05</b>	10.35	8.44	2.89
	WS model (3)	7.84	<b>1.66</b>	1.74	10.53	11.56	3.55
	WS model (4)	6.99	1.79	<b>0.83</b>	10.56	11.92	3.86
	WS model (5)	6.52	1.74	<b>1.49</b>	10.73	12.29	3.07
	Dolphins	6.98	7.32	5.42	10.10	7.73	<b>4.85</b>

**Table 7** continued

$\beta$	Network	Gauss	GSSI	TRBS	SNF	OSBFS	OSDIS
	Lesmis	7.27	5.78	4.65	10.61	5.81	<b>4.29</b>
	PDZBase	10.69	5.13	5.39	13.12	9.59	<b>4.82</b>
	USAirlines	7.31	<b>4.40</b>	4.91	7.21	7.96	5.53
	NetScience	11.81	5.35	5.82	9.47	7.22	<b>4.98</b>
	Celegans	6.92	3.72	<b>3.52</b>	6.76	7.87	4.34
	Euroroads	23.77	7.31	9.32	28.66	14.91	<b>6.67</b>

The values with bold font are the optimal values

by the following function.

$$\begin{aligned} \hat{t^*} &= \arg \min \sum_{k=1}^{\mathcal{K}} \left( t_{v_{o_k}} - t_{o_k} \right)^2 \\ &= \arg \min \sum_{k=1}^{\mathcal{K}} \left( t^* + \sum_{j \in p(s, v_{o_k})} \theta_j - t_{o_k} \right)^2 \end{aligned} \quad (5)$$

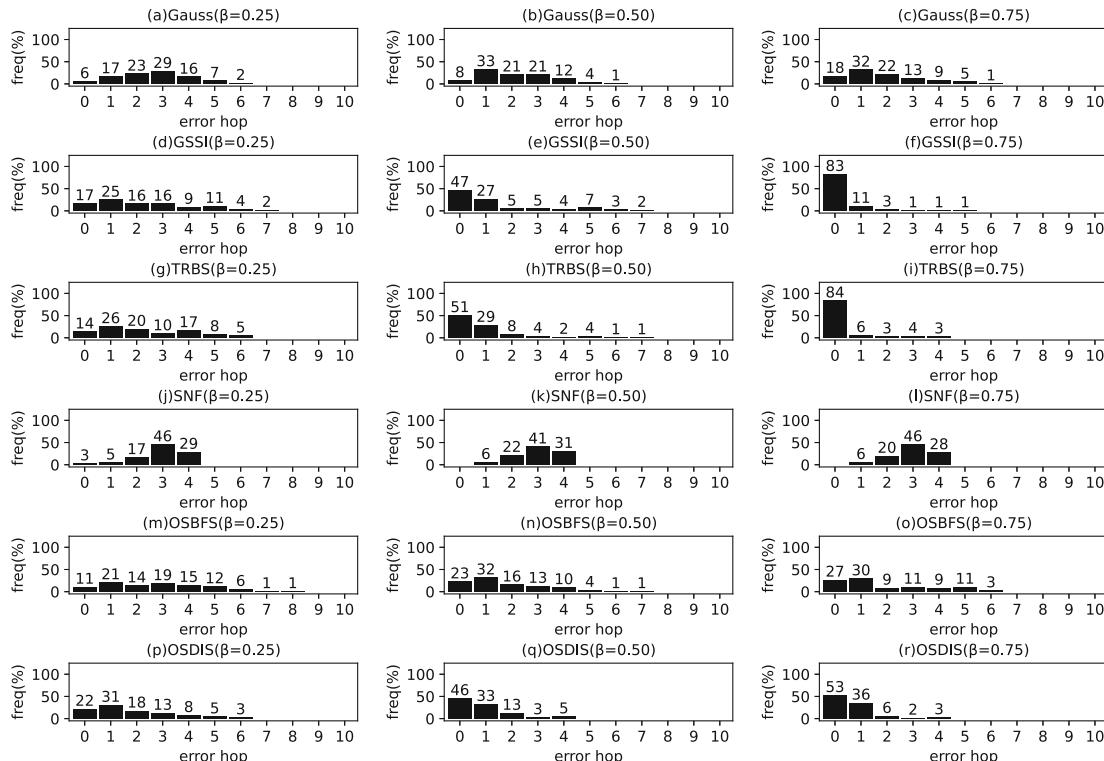
where  $t^* \in [0, z]$ ,  $z \leq \sum \theta_j$ ,  $\theta \in \boldsymbol{\theta}$ .  $t_{v_{o_k}}$  denotes the Measuring Infection Time of  $v_{o_k}$ .  $t_{o_k}$  denotes the Infection Timing information recorded of  $o_k$ .

**Definition 3** Infection Time Similarity is defined as:

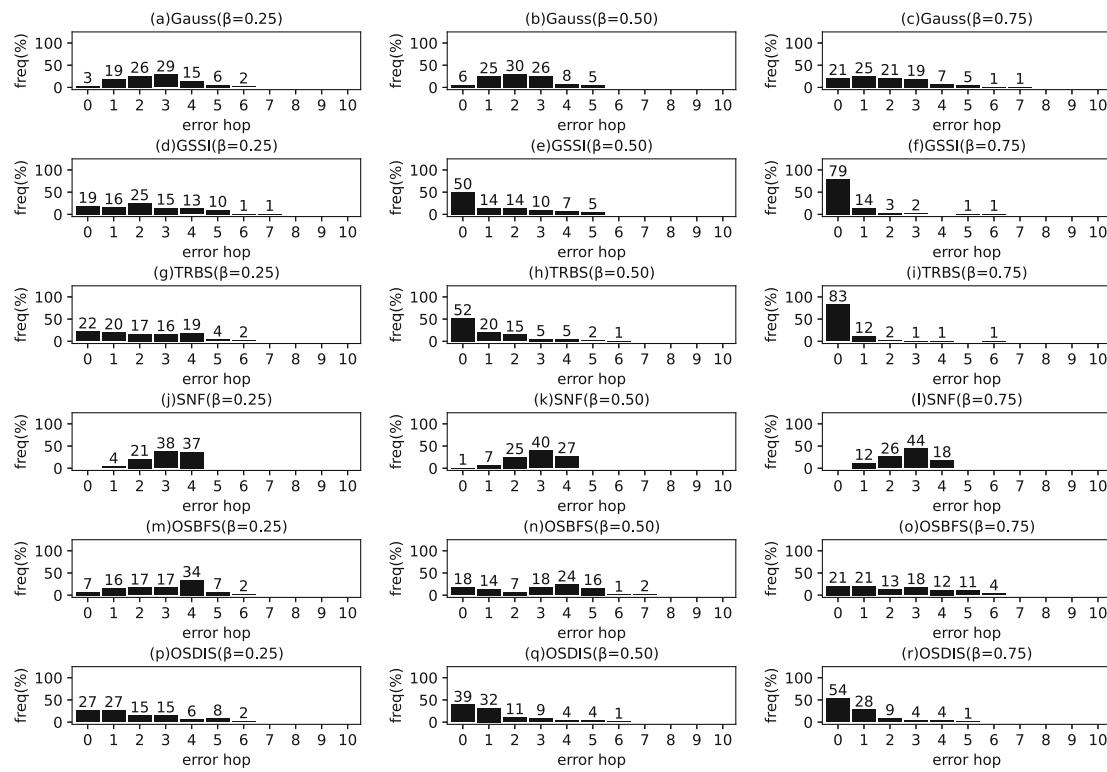
$$\begin{aligned} \mathcal{S}(\mathbf{T}_{\mathcal{O}}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) &= \frac{1}{1 + \mathcal{D}(\mathbf{T}_{\mathcal{O}}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}})} \\ &= \frac{1}{1 + \left( \sum_{k=1}^{\mathcal{K}} |t_{v_{o_k}} - t_{o_k}|^2 \right)^{1/2}} \end{aligned} \quad (6)$$

where  $\mathcal{D}(\mathbf{T}_{\mathcal{O}}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}})$  denotes the Euclidean distance (Rui and Wunsch 2005) between the  $\mathbf{T}_{\mathcal{O}}$  (Definition 1) and the  $\mathbf{T}_{\mathcal{T}_{\text{DIS},s}}$  (Definition 2).  $\mathcal{S}(\mathbf{T}_{\mathcal{O}}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) \in (0, 1]$ .

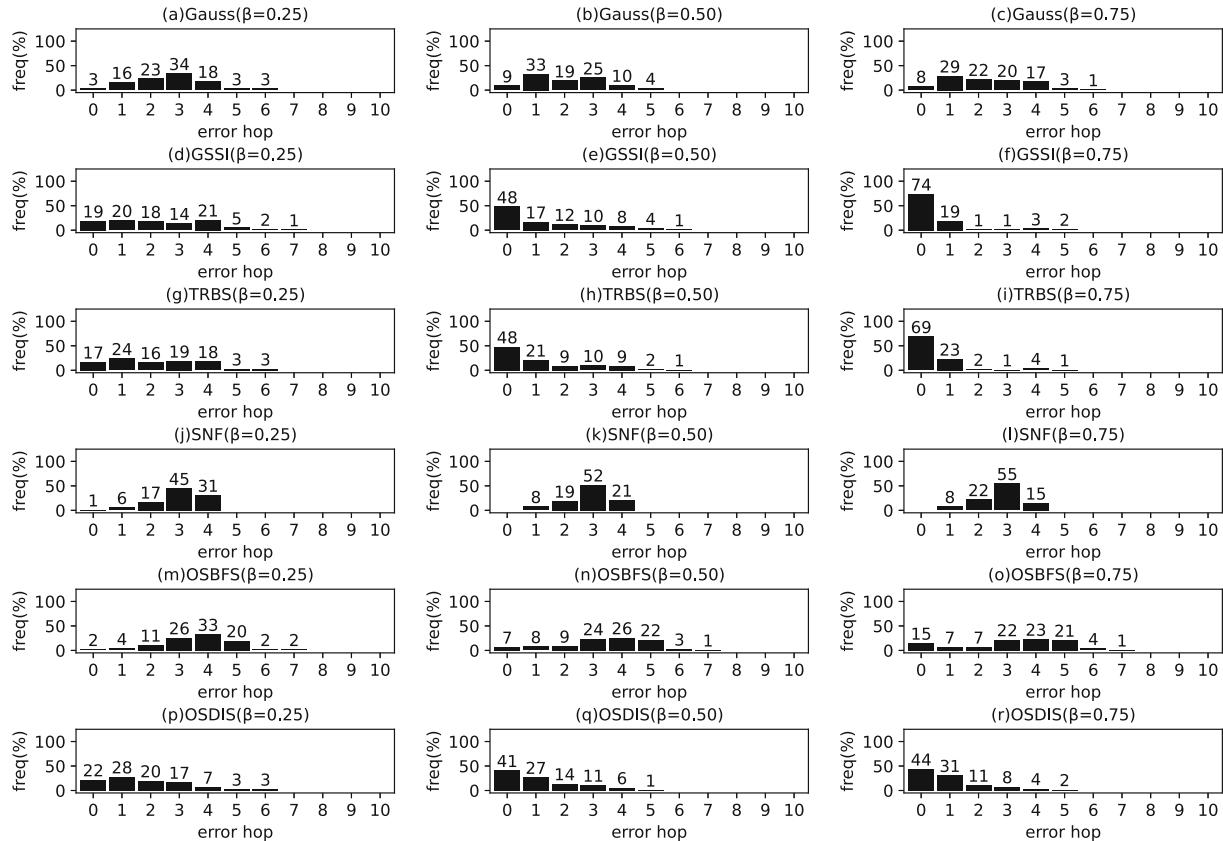
**Definition 4** Observation Infection Time Order. Given an arbitrary  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{\theta})$  and an observers set  $\mathcal{O} = \{o_k\}_{k=1}^{\mathcal{K}}$ . The Observation Time Order of  $\mathcal{O}$  is defined as an ordered observers sequence  $\mathbf{TO}_{\mathcal{O}} = \langle o_i \rangle_{i=1}^{\mathcal{K}}$ , in which each  $o_i$  is



**Fig. 9** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on WS model (1). Each sub-figure is obtained by 100 runs



**Fig. 10** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on WS model (2). Each sub-figure is obtained by 100 runs



**Fig. 11** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on WS model (3). Each sub-figure is obtained by 100 runs

sorted by ascending order according to the Infection Timing information (denoted by  $t_{o_i}$ ) recorded in  $o_i$ . For any pair of observers  $o_i, o_{i+1} \in \mathbf{TO}_{\mathcal{O}}$ , there is  $t_{o_i} \leq t_{o_{i+1}}$ .

**Definition 5** Measuring Infection Time Order. Given an arbitrary  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \theta)$  and an observers set  $\mathcal{O} = \{o_k\}_{k=1}^{\mathcal{K}}$ . The Measuring Infection Time Order of  $\mathcal{O}$  on  $\mathcal{T}_{\text{DIS},s}$  is defined as an ordered nodes sequence  $\mathbf{TO}_{\mathcal{T}_{\text{DIS},s}} = (v_{o_k})_{k=1}^{\mathcal{K}}$ , in which each  $v_{o_k} \in \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}}$  is sorted by ascending order according to the Measuring Infection Time  $t_{v_{o_k}}$  (Definition 2). For any pair of nodes  $v_{o_k}, v_{o_{k+1}} \in \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}}$ , there is  $t_{v_{o_k}} \leq t_{v_{o_{k+1}}}$ .

**Definition 6** Infection Time Order Similarity is defined as:

$$\mathcal{S}(\mathbf{TO}_{\mathcal{O}}, \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}}) = \frac{1 + \tau(\mathbf{TO}_{\mathcal{O}}, \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}})}{2} \quad (7)$$

where  $\tau$  denotes the correlation coefficient defined in reference (Kendall 1938); the details can be found in Appendix A.  $\tau$  is mainly used to measure the concordance between the  $\mathbf{TO}_{\mathcal{O}}$  (Definition 4) and the  $\mathbf{TO}_{\mathcal{T}_{\text{DIS},s}}$  (Definition 5).  $\mathcal{S}(\mathbf{TO}_{\mathcal{O}}, \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}}) \in [0, 1]$ .

The properties related to the Infection Time Similarity (Definition 3) and the Infection Time Order Similarity (Definition 6) can be found in Appendix B.

*Example:* In Fig. 1, Fig. 1b shows a relaxed DIS spanning tree of the network shown in Fig. 1a.  $\mathcal{O} = \{o_1, o_2, o_3\}$ ,  $o_1, o_2$  and  $o_3$  correspond to nodes 2, 3 and 6, respectively. According to Definition 1, for  $\mathbf{T}_{\mathcal{O}}$ ,  $t_2 = 4, t_3 = 7, t_6 = 6$ . When the current root is node 1, according to Eq. 5,  $t^* = 1$ . According to Definition 2, for  $\mathbf{T}_{\mathcal{T}_{\text{DIS},s}}$ ,  $t_2 = 4, t_3 = 7, t_6 = 6$ . Then, according to Definition 3, we have  $\mathcal{S}(\mathbf{T}_{\mathcal{O}}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) = 1$ . According to Definition 4,  $\mathbf{TO}_{\mathcal{O}} = \langle 2, 6, 3 \rangle$ . According to Definition 5,  $\mathbf{TO}_{\mathcal{T}_{\text{DIS},s}} = \langle 2, 6, 3 \rangle$ . Further, with Definition 6, we have  $\mathcal{S}(\mathbf{TO}_{\mathcal{O}}, \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}}) = 1$ .

## 4.2 Locating the propagation source

By combining the Infection Time Similarity (Definition 3) and the Infection Time Order Similarity (Definition 6), the source estimator in Eq. 2 can be written as follows:

$$\begin{aligned} \hat{s} &= \arg \max_{s \in \mathcal{V}} \mathcal{S}(\mathcal{O}_{T_{s^*}}, \mathcal{O}_{\mathcal{T}_{\text{DIS},s}}) \\ &= \arg \max_{s \in \mathcal{V}} (\mathcal{S}(\mathbf{T}_{\mathcal{O}}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) \times \mathcal{S}(\mathbf{TO}_{\mathcal{O}}, \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}})) \end{aligned} \quad (8)$$

where  $\mathcal{S}(\mathbf{T}_{\mathcal{O}}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}})$  and  $\mathcal{S}(\mathbf{TO}_{\mathcal{O}}, \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}})$  are defined in Definition 3 and Definition 6, respectively.

Based on Eq. 8, we propose a novel source locating method, termed as OSDIS algorithm, which is summarized in Algorithm 2.

**Algorithm 2** analysis: The  $E(\mathcal{T})$  declared in line 2 is an edge set. Lines 3–19 are used for traversing the  $\mathcal{G}$  by the relaxed DIS with current node  $s$  as root and recording the eligible edges into  $E(\mathcal{T})$ . Lines 3–19 require  $O(|\mathcal{V}|^2)$  computations in the worst case. Line 20 requires  $O(|\mathcal{V}| + |\mathcal{E}|)$  computations, which can be reduced to  $O(|\mathcal{E}|)$ . In lines 21–22, the  $\mathcal{T}$  obtained in line 20 will be marked as a relaxed DIS spanning tree  $\mathcal{T}_{\text{DIS},s}$  if and only if  $|E(\mathcal{T})| == |\mathcal{V}| - 1$ . Line 23 requires  $O(\mathcal{K})$  computations. Line 24 requires  $O(|\mathcal{V}|^2 + z|\mathcal{V}|\mathcal{K})$  computations ( $z$  can be found in Eq. 5). Lines 25–26 require  $O(|\mathcal{V}|)$  and  $O(\mathcal{K}^2)$  computations, respectively. Both lines 27 and 28 require  $O(\mathcal{K} \log \mathcal{K})$  computations. Line 29 requires  $O(\mathcal{K}^2)$  computations. Finally, each node  $s \in \mathcal{V}$  will be used as root to construct different  $\mathcal{T}_{\text{DIS},s}$ . Thus, the time complexity of Algorithm 2 is  $O(|\mathcal{V}|^3 + z|\mathcal{V}|^2\mathcal{K})$ .

---

### Algorithm 2 OSDIS algorithm

---

**Input:** an arbitrary  $\mathcal{G}$  and  $\mathcal{O}$

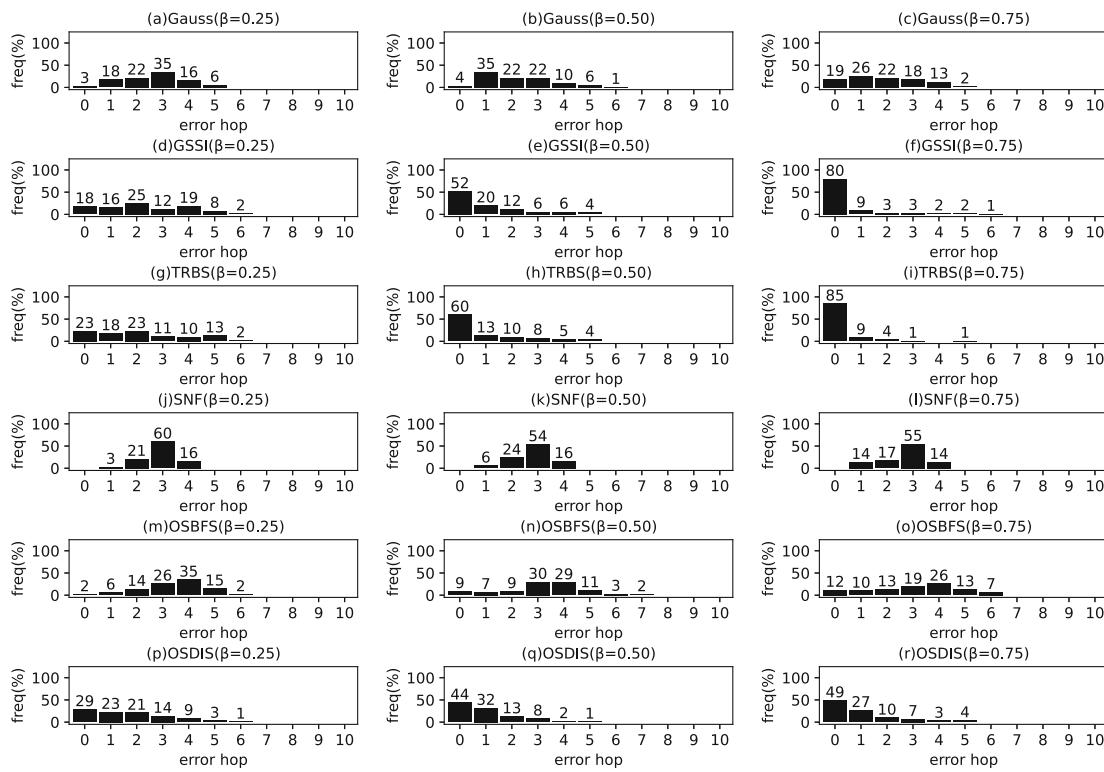
**Output:** the estimated propagation source  $\hat{s}$

```

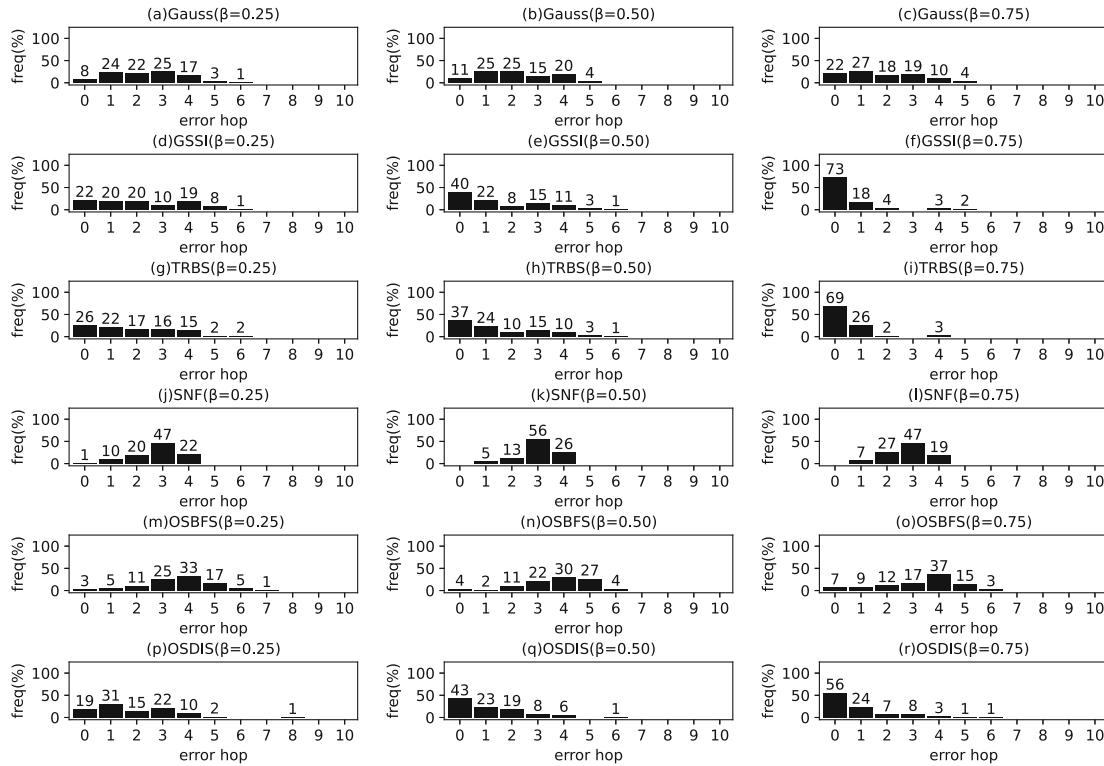
1: for each node  $s \in \mathcal{V}$  do
2:   initialize an empty set  $E(\mathcal{T})$ .
3:   initialize an empty queue  $\mathcal{Q}$ .
4:    $\mathcal{Q}.\text{enqueue}(s)$  and set  $s$  as visited.
5:   while  $\mathcal{Q}$  is not empty do
6:      $v = \mathcal{Q}.\text{front}()$ .
7:      $\mathcal{Q}.\text{dequeue}()$ .
8:     for each neighbor  $u$  of  $v$  do
9:       if  $u$  has not been visited then
10:        if  $u \in \mathcal{O}$  then
11:          if the Diffusion Direction information recorded in  $u$  is  $v$  then
12:             $\mathcal{Q}.\text{enqueue}(u)$ , set  $u$  as visited,  $E(\mathcal{T}) = E(\mathcal{T}) \cup e_{v \rightarrow u}$ .
13:          end if
14:        else
15:           $\mathcal{Q}.\text{enqueue}(u)$ , set  $u$  as visited,  $E(\mathcal{T}) = E(\mathcal{T}) \cup e_{vu}$ .
16:        end if
17:      end if
18:    end for
19:  end while
20:  remove the edges  $\notin E(\mathcal{T})$  from  $\mathcal{G}$  and mark the remaining part of  $\mathcal{G}$  as  $\mathcal{T}$ .
21:  if  $|E(\mathcal{T})| == |\mathcal{V}| - 1$  then
22:    mark  $\mathcal{T}$  as relaxed DIS spanning tree  $\mathcal{T}_{\text{DIS},s}$ .
23:    obtain  $\mathbf{T}_{\mathcal{O}}$  according to Definition 1.
24:    estimate  $t^*$  according to equation 5.
25:    compute  $\mathbf{T}_{\mathcal{T}_{\text{DIS},s}}$  according to Definition 2.
26:    compute  $\mathcal{S}(\mathbf{T}_{\mathcal{O}}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}})$  according to equation 6.
27:    compute  $\mathbf{TO}_{\mathcal{O}}$  according to Definition 4.
28:    compute  $\mathbf{TO}_{\mathcal{T}_{\text{DIS},s}}$  according to Definition 5.
29:    compute  $\mathcal{S}(\mathbf{TO}_{\mathcal{O}}, \mathbf{TO}_{\mathcal{T}_{\text{DIS},s}})$  according to equation 7.
30:    compute  $\mathcal{S}(\mathcal{O}_{T_{s^*}}, \mathcal{O}_{\mathcal{T}_{\text{DIS},s}})$  according to equation 8.
31:  end if
32: end for
33: pick  $\hat{s}$  with the maximal value according to equation 8.

```

---

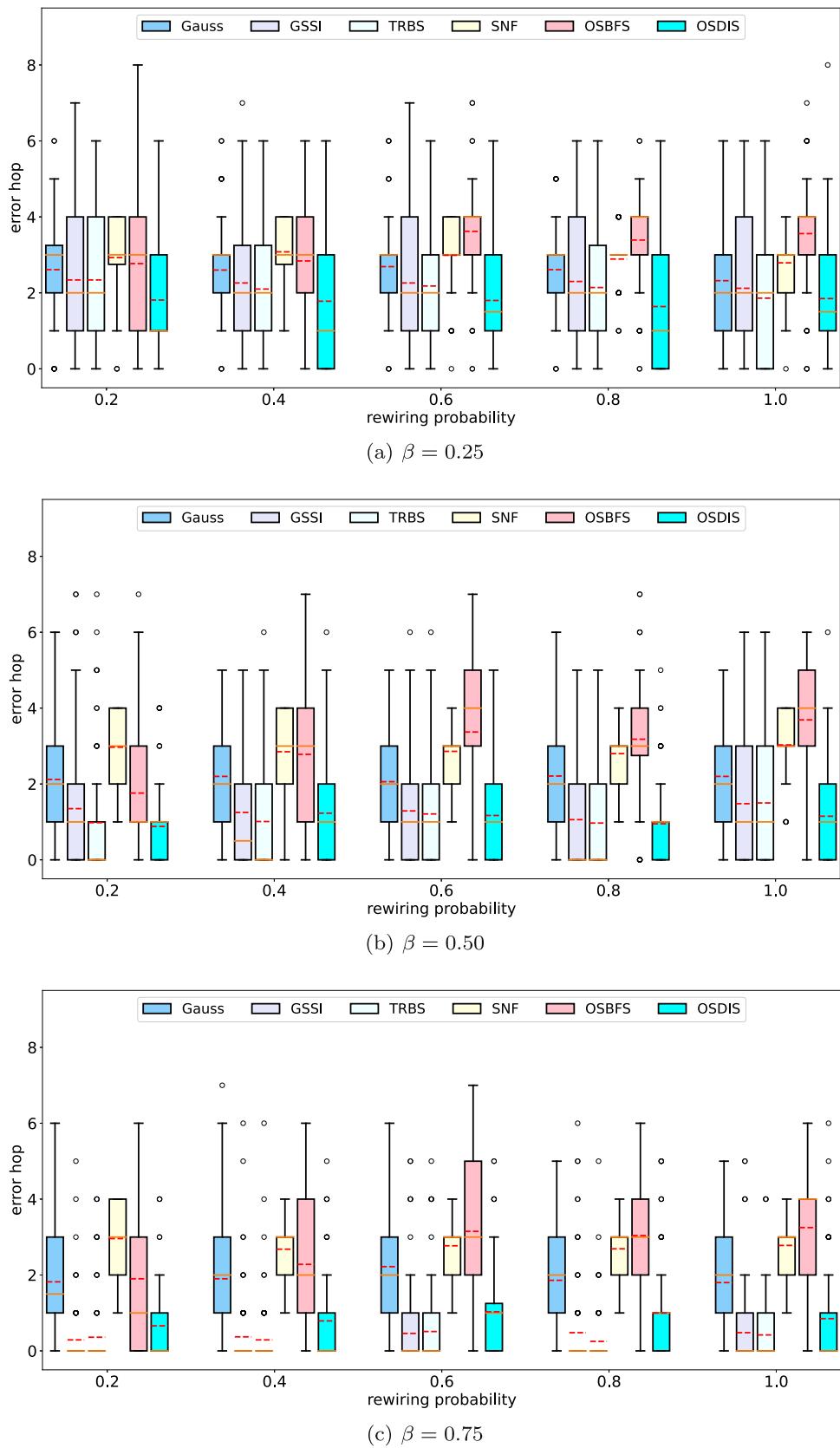


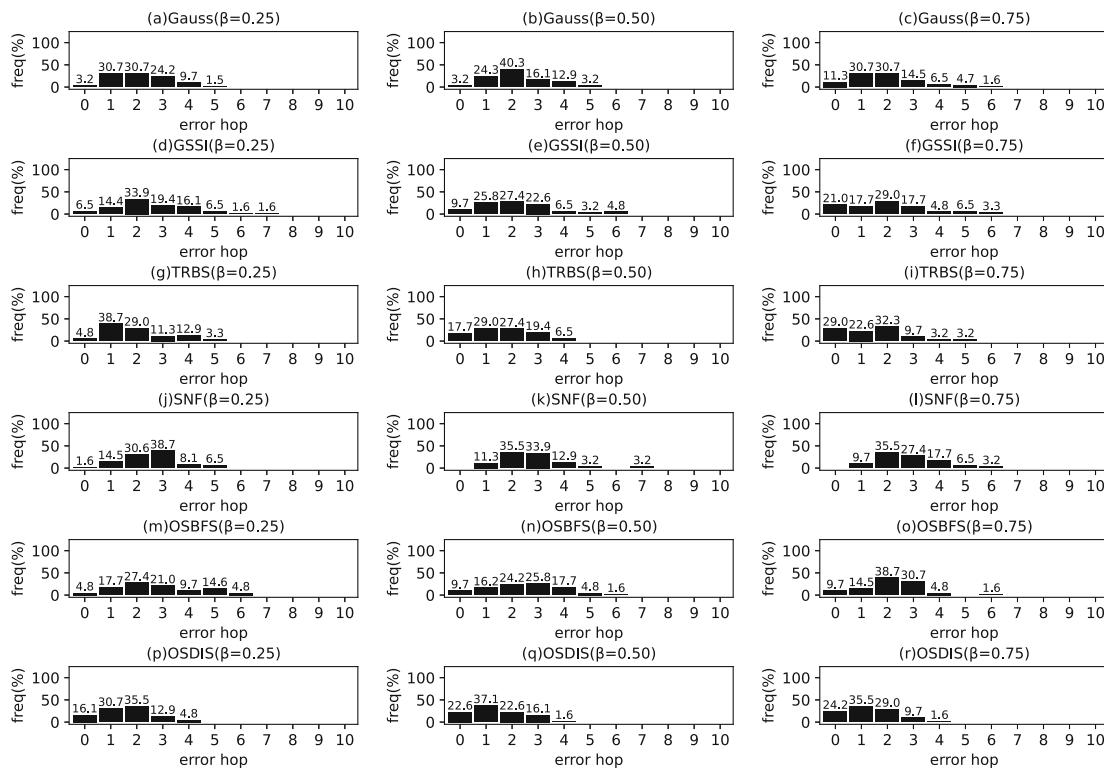
**Fig. 12** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on WS model (4). Each sub-figure is obtained by 100 runs



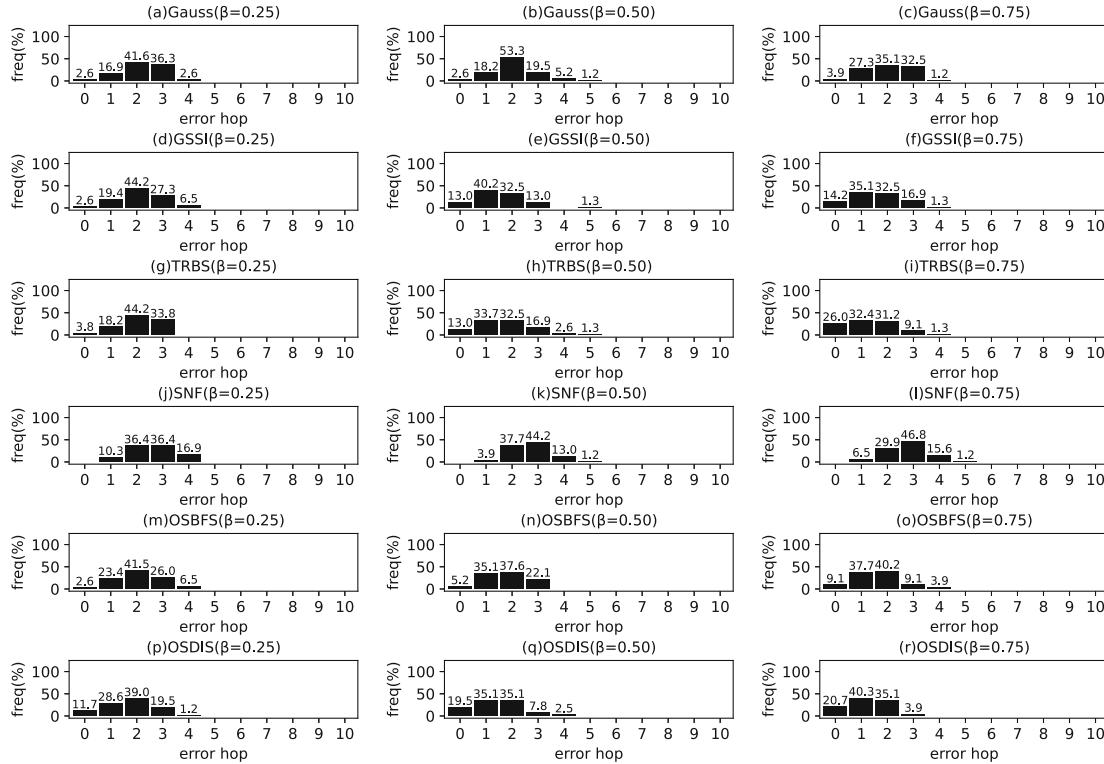
**Fig. 13** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on WS model (5). Each sub-figure is obtained by 100 runs

**Fig. 14** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on the WS models generated by different rewiring probabilities. The orange line and red line in each box denote the median and average error hop (also shown in Table 6), respectively (color figure online)

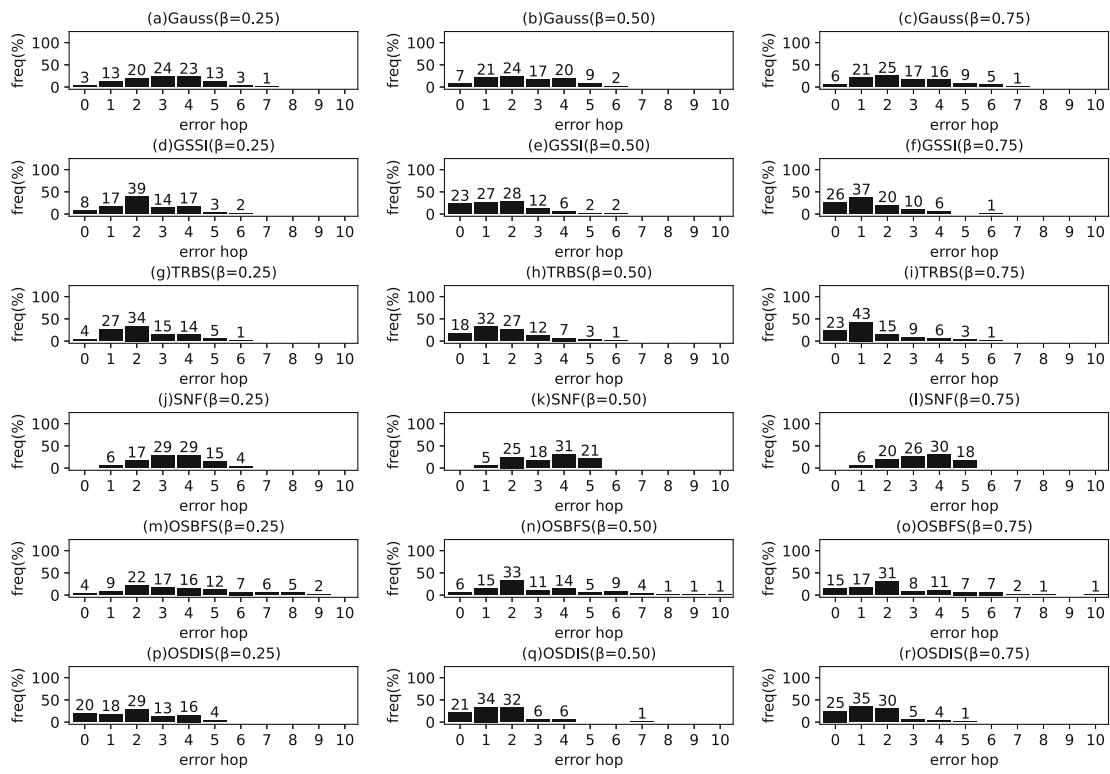




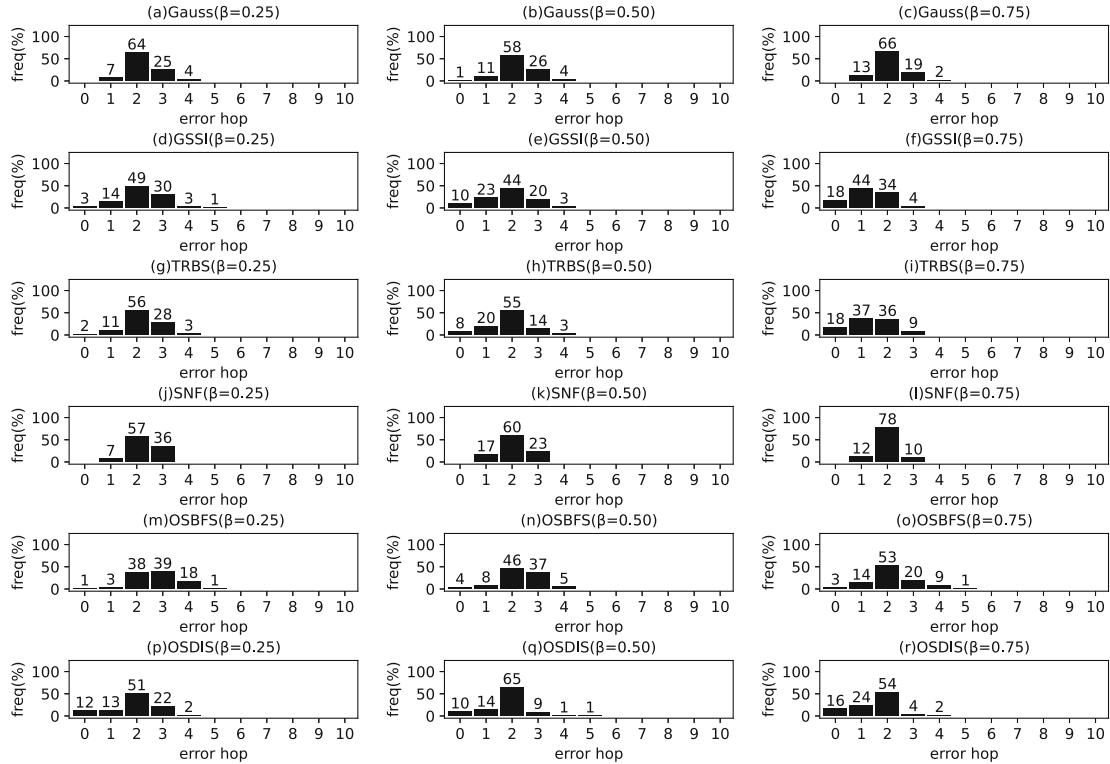
**Fig. 15** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on Dolphins network. Each sub-figure is obtained by 62 runs



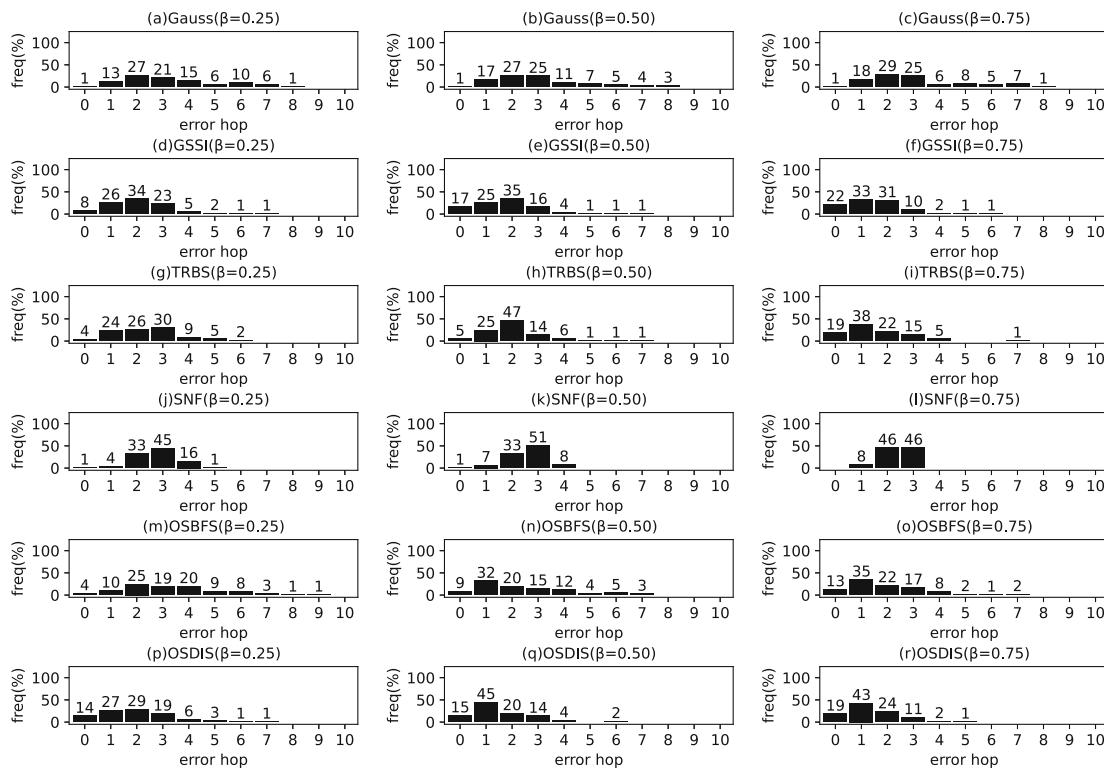
**Fig. 16** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on Lesmis network. Each sub-figure is obtained by 77 runs



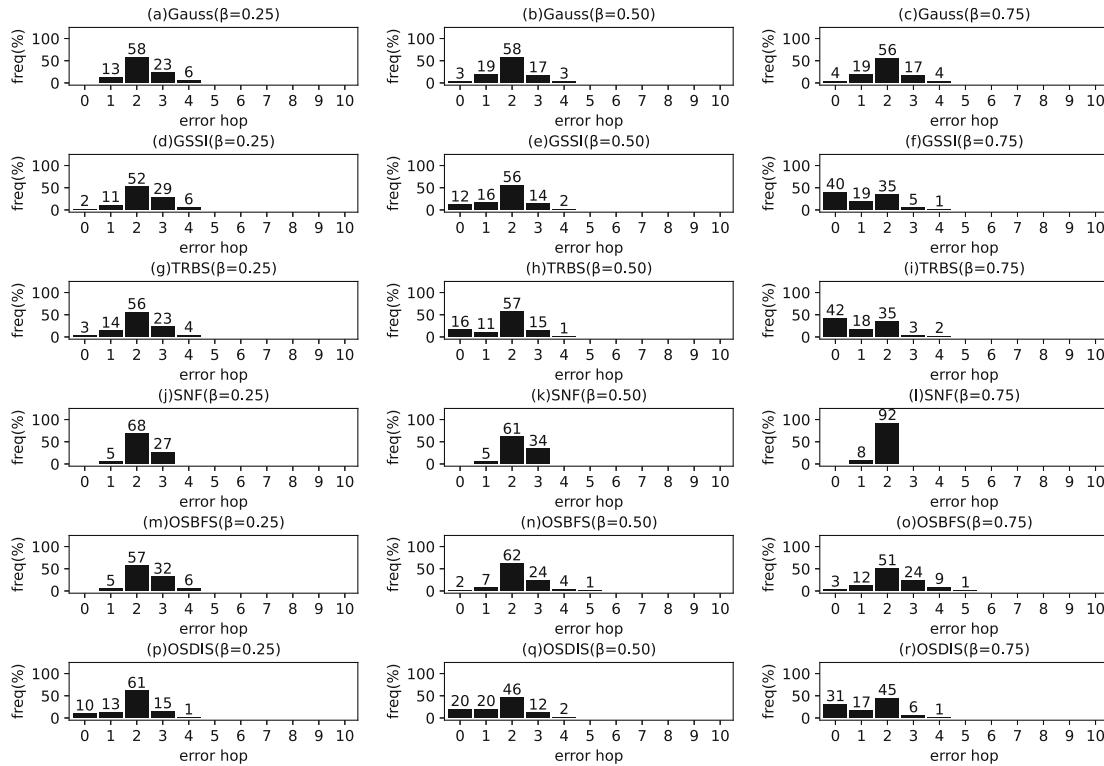
**Fig. 17** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on PDZBase network. Each sub-figure is obtained by 100 runs



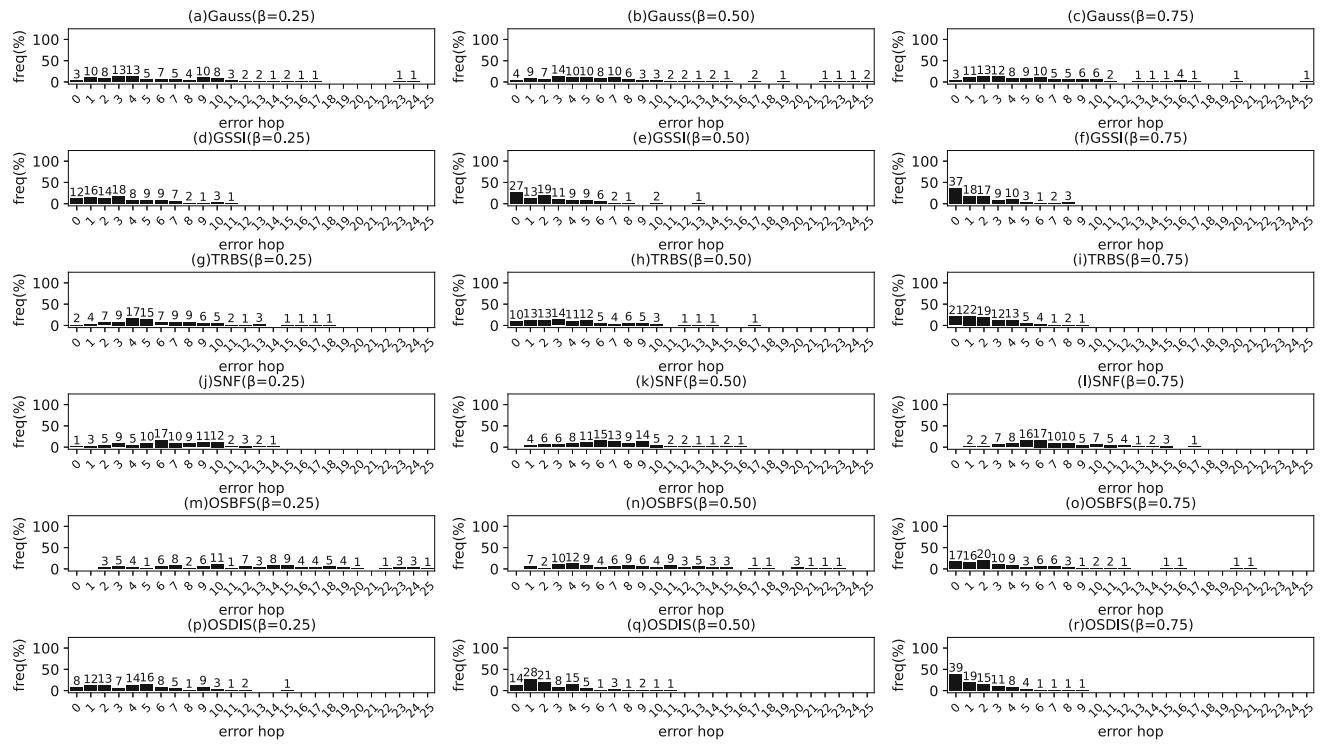
**Fig. 18** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on USAirlines network. Each sub-figure is obtained by 100 runs



**Fig. 19** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on NetScience network. Each sub-figure is obtained by 100 runs



**Fig. 20** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on C elegans network. Each sub-figure is obtained by 100 runs



**Fig. 21** The results of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms applied on Euroroads network. Each sub-figure is obtained by 100 runs

## 5 Experimental evaluation

To validate the feasibility and effectiveness of the OSDIS algorithm, it is compared with other four state-of-the-art methods on a series of synthetic and real networks. The four methods include the Gauss algorithm (Pinto et al. 2012), GSSI algorithm (Tang et al. 2018), TRBS algorithm (Shen et al. 2016) and SNF algorithm (Wang and Sun 2020). Besides, since the OSDIS algorithm is based on the relaxed DIS heuristic, to show its advantage, we also define an algorithm, denoted by OSBFS, in which the relaxed DIS heuristic is replaced by the breadth-first search (BFS) heuristic. Totally, six algorithms are compared in the experiments. Their time complexity is shown in Table 2. Similar to the reference (Yang et al. 2020), the performance of a source locating algorithm is mainly evaluated by the precision (the precise locating ratio, i.e., the proportion of 0 error hop), the average error hop and the average error delay. For the precision, the higher the value is, the better the algorithm is. For the average error hop and the average error delay, the smaller the value is, the better the algorithm is.

**Running environment** Hardware: Dell R740 with 2 Intel® Xeon® gold 6254 CPU, 1T RAM. Software: Cygwin 3.0.7 + Eclipse Cpp2019 + igraph C 0.7.1 + Eigen/Dense (used for running algorithms). R 64 × 3.3.3 + igraph R 1.2.1 (used for generating synthetic networks).

**Datasets** The six algorithms are evaluated on a series of synthetic and real networks. The synthetic networks include the scale-free (BA) model (Barabasi and Albert 1999) and the small-world (WS) model (Watts and Strogatz 1998). Totally, six BA models with different powers of the preferential attachment and five WS models with different rewiring probabilities are generated, respectively. The detailed parameters for generating these synthetic networks are shown in Table 3. The real networks are selected from different fields, which can be obtained from the Koblenz Network Collection (Kunegis 2013) and the Network Data Repository (Rossi and Ahmed 2015) for free. All the real networks are shown in Table 4. The topology properties of the used networks are shown in Table 5.

**Parameters setting** Given an arbitrary graph  $\mathcal{G}$ , the propagation delays set  $\theta$  are independent identically distributed (i.i.d) random variables with Gaussian distribution  $\mathcal{N}(\mu, \sigma^2)$ ,  $\mu$  and  $\sigma^2$  are known (Pinto et al. 2012; Paluch et al. 2018). We set  $\mu/\sigma = 4$ . The diffusion model follows the one introduced in Sect. 3. To investigate the impact of different propagation ratios (denoted by  $\beta$ ) on the performance of the source locating algorithms, we set  $\beta = 0.25$ ,  $\beta = 0.50$  and  $\beta = 0.75$ , respectively. Additionally, to compare with the GSSI algorithm (Tang et al. 2018), we set  $s^* \notin \mathcal{O}$ . Generally, in reality, to save the cost, the number of the observers will be far less than the size of  $\mathcal{G}$ . Thus, we randomly select 5% nodes as the observers in each network.

**Table 8** The average running time ratios between other five algorithms and the OSDIS on different networks

Network	$\frac{t(\text{Gauss})}{t(\text{OSDIS})}$	$\frac{t(\text{GSSI})}{t(\text{OSDIS})}$	$\frac{t(\text{TRBS})}{t(\text{OSDIS})}$	$\frac{t(\text{SNF})}{t(\text{OSDIS})}$	$\frac{t(\text{OSBFS})}{t(\text{OSDIS})}$
BA model (1)	59.52	10.27	0.0004	0.03	0.68
BA model (2)	52.24	9.61	0.0003	0.02	0.58
BA model (3)	55.94	9.77	0.0004	0.02	0.39
BA model (4)	54.16	9.62	0.0004	0.02	0.31
BA model (5)	56.71	9.84	0.0006	0.02	0.24
BA model (6)	55.22	9.83	0.0006	0.02	0.24
WS model (1)	43.98	7.56	0.0007	0.03	1.04
WS model (2)	54.13	9.29	0.0004	0.03	1.00
WS model (3)	57.22	10.00	0.0005	0.04	0.86
WS model (4)	59.48	10.34	0.0003	0.04	0.83
WS model (5)	61.03	10.50	0.0003	0.06	0.80
Dolphins	42.52	16.81	0.0067	0.29	0.90
Lesmis	30.69	9.52	0.0100	0.11	0.83
PDZBase	60.84	11.67	0.0007	0.05	0.60
USAirlines	75.98	11.71	0.0002	0.02	0.68
NetScience	104.62	15.35	0.0001	0.02	1.18
Celegans	138.81	19.98	0.0001	0.01	0.69
Euroroads	1342.74	180.82	$\approx 0$	0.02	1.06

$t(\cdot)$ : the running time of an algorithm

## 5.1 Experimental results on the synthetic networks

Figures 2, 3, 4, 5, 6, 7 show the precision (the precise locating ratio, i.e., the proportion of 0 error hop) of the six algorithms on a series of BA models. From Figs. 2, 3, 4, 5, 6, 7, we can see that, when  $\beta = 0.25$ ,  $\beta = 0.5$  and  $\beta = 0.75$ , the OSDIS algorithm generally exposes the best precision on all the six BA models, i.e., the OSDIS has a higher proportion in 0 error hop than other five algorithms. Only when  $\beta = 0.75$ , the OSDIS is inferior to the GSSI and TRBS on BA model (1), but outperforms other three algorithms. From Table 6, we know that, when  $\beta = 0.25$ ,  $\beta = 0.5$  and  $\beta = 0.75$ , the OSDIS is better than other five algorithms in the average error hop on all the six BA models. Only when  $\beta = 0.5$ , the OSDIS is a litter inferior to OSBFS on BA model (5), but superior to other four algorithms. Meanwhile, in Fig. 8, we plot interquartile range (IQR) to show the distribution regions of error hop of the six algorithms on the BA models. Additionally, from Table 7, we can see that, when  $\beta = 0.25$ ,  $\beta = 0.5$  and  $\beta = 0.75$ , the OSDIS exposes a better average error delay on all the six BA models. Only when  $\beta = 0.75$ , the OSDIS is inferior to TRBS on BA model (1), but outperforms other four algorithms. In summary, on the BA models, the OSDIS is generally better than other five algorithms in the precision, the average error hop and average error delay.

Figures 9, 10, 11, 12, 13 show the precision (the precise locating ratio, i.e., the proportion of 0 error hop) of the six algorithms on a series of WS models. From Figs. 9, 10, 11, 12, 13, we can see that, when  $\beta = 0.25$ , the OSDIS is superior to

other five algorithms in the precision on WS models (1)–(4) (i.e., the OSDIS has a higher proportion in 0 error hop), but only inferior to TRBS and GSSI on WS model (5). When  $\beta = 0.5$  and  $\beta = 0.75$ , the OSDIS is generally inferior to TRBS and GSSI in the precision on WS models (1)–(5), but superior to other three algorithms. Only when  $\beta = 0.5$ , the OSDIS exposes the best precision on WS model (5). From Tables 6 and 7, we know that, when  $\beta = 0.25$  and  $\beta = 0.5$ , the OSDIS is generally better than other five algorithms in the average error hop and average error delay on all the five WS models. Only when  $\beta = 0.5$ , the OSDIS is inferior to TRBS on WS model (2). Meanwhile, from Tables 6 and 7, we know that, when  $\beta = 0.75$ , the OSDIS is always inferior to GSSI and TRBS in the average error hop and average error delay, but outperforms other three algorithms. In Fig. 14, we further plot interquartile range (IQR) to show the distribution regions of error hop of the six algorithms on the WS models. In summary, on the WS models, the OSDIS is generally superior to other five algorithms in the precision when  $\beta = 0.25$  and generally exposes a better performance in the average error hop and average error delay when  $\beta = 0.25$  and  $\beta = 0.5$ . Thus, the OSDIS is better than other five algorithms in most cases. Obviously, the performance of the OSDIS on the BA models is better than on the WS models.

## 5.2 Experimental results on the real networks

In this subsection, we further validate the performance of the six algorithms on the real networks. Figures 15, 16, 17, 18,

[19](#), [20](#), [21](#) show the precision (the precise locating ratio, i.e., the proportion of 0 error hop) of the six algorithms on the real networks. When  $\beta = 0.25$ , from Figs. [15](#), [16](#), [17](#), [18](#), [19](#), [20](#), [21](#), we can see that the OSDIS generally exposes the best precision (i.e., the OSDIS has a higher proportion in 0 error hop) on all the real networks, except for Euroroads network on which the OSDIS is only inferior to GSSI, but superior to other four algorithms. By combining with Tables [6](#) and [7](#), we know that the OSDIS is also superior to other five algorithms in the average error hop and the average error delay on all the real networks, except for Euroroads network on which the OSDIS is only inferior to GSSI, but superior to other four algorithms. When  $\beta = 0.5$ , from Figs. [15](#), [16](#), [17](#), [18](#), [19](#), [20](#), [21](#), we know that the OSDIS exposes the best precision (i.e., the OSDIS has a higher proportion in 0 error hop) on Dolphins, Lesmis, USAirlines and Celegans networks. Meanwhile, the OSDIS is only inferior to GSSI in the precision on PDZBase, NetScience and Euroroads networks, but superior to other four algorithms. By combining with Tables [6](#) and [7](#), we can see that the OSDIS generally outperforms other five algorithms in the average error hop and the average error delay on the real networks. Only on USAirlines network, the OSDIS is inferior to GSSI and TRBS in the average error delay. When  $\beta = 0.75$ , from Figs. [15](#), [16](#), [17](#), [18](#), [19](#), [20](#), [21](#), we know that, except for Euroroads network, the OSDIS is generally inferior to TRBS or GSSI in the precision on the real networks. By combining with Tables [6](#) and [7](#), we can see that the OSDIS outperforms other five algorithms in the average error hop and average error delay on Dolphins, Lesmis, PDZBase, NetScience and Euroroads networks, but is inferior to GSSI and TRBS on USAirlines and Celegans networks. Meanwhile, in Appendix C Figs. [22](#), [23](#), [24](#), [25](#), [26](#), [27](#), [28](#), we plot interquartile range (IQR) to further show the distribution regions of error hop of the six algorithms on the real networks. In summary, on the real networks, the OSDIS is generally superior to other five algorithms in the precision, the average error hop and the average error delay when  $\beta = 0.25$  and  $\beta = 0.5$ , but inferior to GSSI and TRBS when  $\beta = 0.75$ . Thus, the OSDIS is better than other five algorithms in most cases.

Overall, in the precision, the average error hop and average error delay, the OSDIS outperforms other five algorithms on the BA models and in most cases is superior to other five algorithms on the WS models and real networks. In a few cases, the OSDIS is only inferior to GSSI and TRBS, but superior to other three algorithms. Thus, the OSDIS is a feasible and effective method in accurately locating the propagation source. Meanwhile, on the BA models, WS models and real networks, the OSDIS obviously outperforms the OSBFS, which indicates that the relaxed DIS heuristic outperforms the BFS heuristic in locating the propagation source.

The average error hops of the six algorithms on different networks are shown in Table [6](#). The average error delay is

shown in Table [7](#). The average running time ratios between other five algorithms and the OSDIS on all networks are shown in Table [8](#). From Table [8](#), we can see that the efficiency of the OSDIS is inferior to the TRBS and SNF, similar with the OSBFS, and superior to the Gauss and GSSI.

## 6 Conclusion

In this paper, we locate the propagation source by utilizing both of the diffusion direction information and the infection time information of the observers. We introduce a relaxed direction-induced search (DIS) to utilize the diffusion direction information of the observers to approximate the actual diffusion tree on a network. Based on the relaxed DIS, we further utilize the infection time information of the observers to define two kinds of observer-based similarity measures, including the Infection Time Similarity and the Infection Time Order Similarity. With the two kinds of similarity measures and the relaxed DIS, a source locating method termed as OSDIS is proposed. The feasibility and effectiveness of the OSDIS are validated on a series of synthetic and real networks. Meanwhile, the experimental results also show that the relaxed DIS heuristic outperforms the BFS heuristic in propagation source locating. The current OSDIS is only developed for single source locating. In the future work, we will study the OSDIS-based multi-sources locating method.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China (Grant No. 62062010), the Science and Technology Planning Project of Guangxi (Grant No. AD19245101), the Science and Technology Planning Project of Liuzhou City (Grant No. 2020PAAA0606), the National Natural Science Foundation of China (Grant No. 62061003, 61963006) and the Doctoral Foundation of Guangxi University of Science and Technology (Grant No. 19Z06).

**Data availability** Enquiries about data availability should be directed to the authors.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copy-

right holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## A. Correlation coefficient

The correlation coefficient  $\tau$  defined in reference (Kendall 1938) considers a set of joint ranks from two sequence  $\mathbf{X}$  and  $\mathbf{Y}$ . For arbitrary  $x_i, x_j \in \mathbf{X}, y_i, y_j \in \mathbf{Y}$ , any pair of two-tuples  $(x_i, y_i)$  and  $(x_j, y_j)$  ( $i \neq j$ ) are said to be concordant if both  $x_i > x_j$  and  $y_i > y_j$  or if both  $x_i < x_j$  and  $y_i < y_j$ . They are said to be discordant if both  $x_i > x_j$  and  $y_i < y_j$  or if both  $x_i < x_j$  and  $y_i > y_j$ . If  $x_i = x_j$  or  $y_i = y_j$ , the pair is neither concordant nor discordant.  $\tau$  is defined as:

$$\tau = \frac{2 \times (n_c - n_d)}{n(n-1)} \quad (9)$$

where  $n_c$  and  $n_d$  denote the number of concordant and discordant pairs, respectively.  $\tau \in [-1, 1]$ .

## B. Properties

**Property 1** Suppose that the propagation ratio of the SI model is  $\beta = 1$ , for  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \theta)$  and  $\mathcal{O} = \{o_k\}_{k=1}^K$ , suppose  $\mathcal{O} = \mathcal{V}$ , then

$$\mathcal{S}(\mathbf{T}_\mathcal{O}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) = 1 \quad (10)$$

**Proof** By Eq. 6, the proof of Eq. 10 can be converted into the proof about  $\mathcal{D}(\mathbf{T}_\mathcal{O}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) = 0$ . Firstly, since  $\beta = 1$ , on the original  $\mathcal{G}$ , each node  $o_k$  is first time infected along the weighted shortest path between  $s^*$  and  $o_k$ . Meanwhile, the

Infection Timing information can be recorded in  $o_k$ , denoted by  $t_{o_k}$  (Definition 1). Secondly, since  $\mathcal{O} = \mathcal{V}$ , by combining with Algorithm 1, we know that the  $\mathcal{T}_{\text{DIS},s}$  is the actual diffusion tree corresponding to the first time each node in  $\mathcal{G}$  gets infected. Thus, for the Measuring Infection Time of each  $v_{o_k}$  (denoted by  $t_{v_{o_k}}$ , Definition 2), there is  $t_{v_{o_k}} = t_{o_k}$ . According to the  $\mathcal{D}(\mathbf{T}_\mathcal{O}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}})$  in Definition 3, there is  $\mathcal{D}(\mathbf{T}_\mathcal{O}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) = 0$ . Property 1 is proved.  $\square$

**Property 2** Suppose that the propagation ratio of the SI model is  $\beta = 1$ , for  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \theta)$  and  $\mathcal{O} = \{o_k\}_{k=1}^K$ , suppose  $\mathcal{O} = \mathcal{V}$ , then

$$\mathcal{S}(\mathbf{T}_\mathcal{O}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) = 1 \quad (11)$$

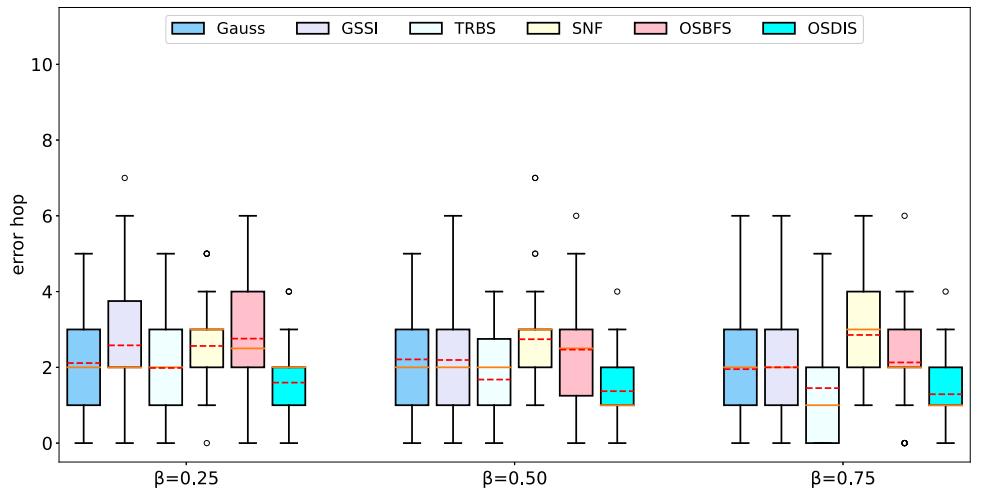
**Proof** By Eq. 7, the proof of Eq. 11 can be converted into the proof about  $\tau(\mathbf{T}_\mathcal{O}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) = 1$ . From Property 1, we know that, since  $\mathcal{O} = \mathcal{V}$ , then we get  $\mathcal{S}(\mathbf{T}_\mathcal{O}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) = 1$ .

For  $\mathbf{T}_\mathcal{O} = \{t_{o_k}\}_{k=1}^K$  and  $\mathbf{T}_{\mathcal{T}_{\text{DIS},s}} = \{t_{v_{o_k}}\}_{k=1}^K$ , there is  $t_{v_{o_k}} = t_{o_k}$ . Thus, the Observation Infection Time Order in  $\mathbf{T}_\mathcal{O}$  is consistent with the Measuring Infection Time Order in  $\mathbf{T}_{\mathcal{T}_{\text{DIS},s}}$ . According to the Definition of correlation coefficient  $\tau$  (Appendix A),  $\tau(\mathbf{T}_\mathcal{O}, \mathbf{T}_{\mathcal{T}_{\text{DIS},s}}) = 1$ . Property 2 is proved.  $\square$

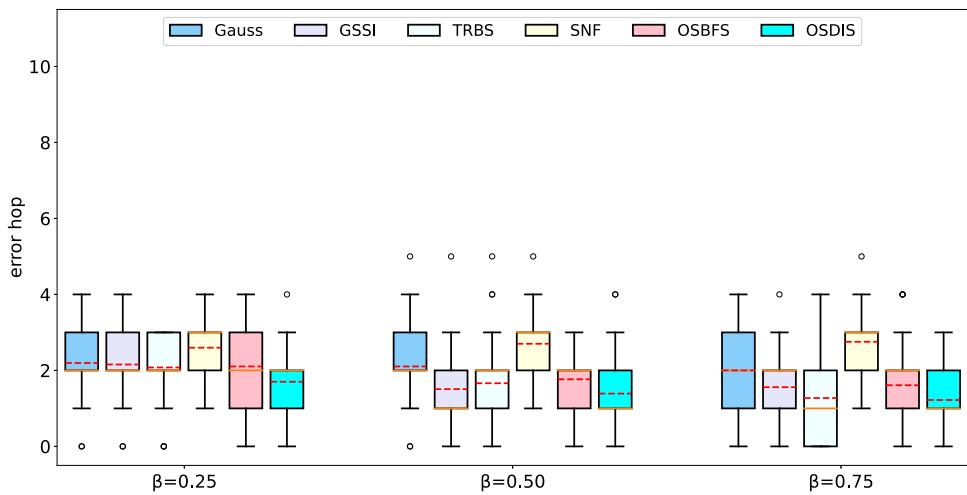
## C. Supplemental material

The interquartile range (IQR) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on the real networks is shown in Figs. 22, 23, 24, 25, 26, 27, 28.

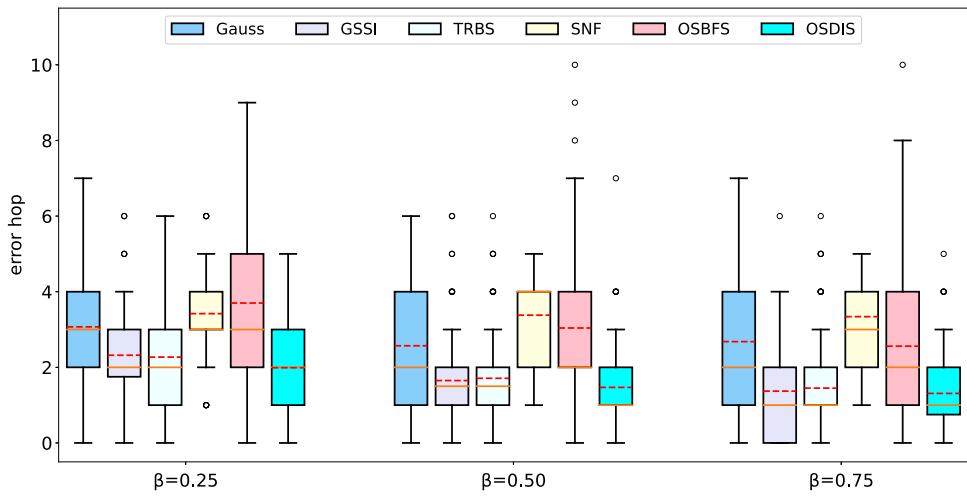
**Fig. 22** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on Dolphins network. The orange line and red line in each box denote the median and average error hop, respectively (color figure online)



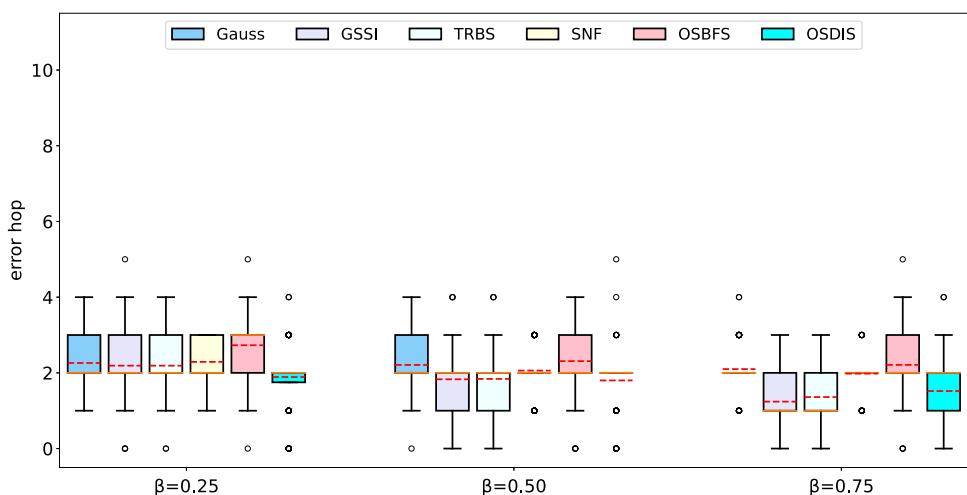
**Fig. 23** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on Lesmis network. The orange line and red line in each box denote the median and average error hop, respectively (color figure online)



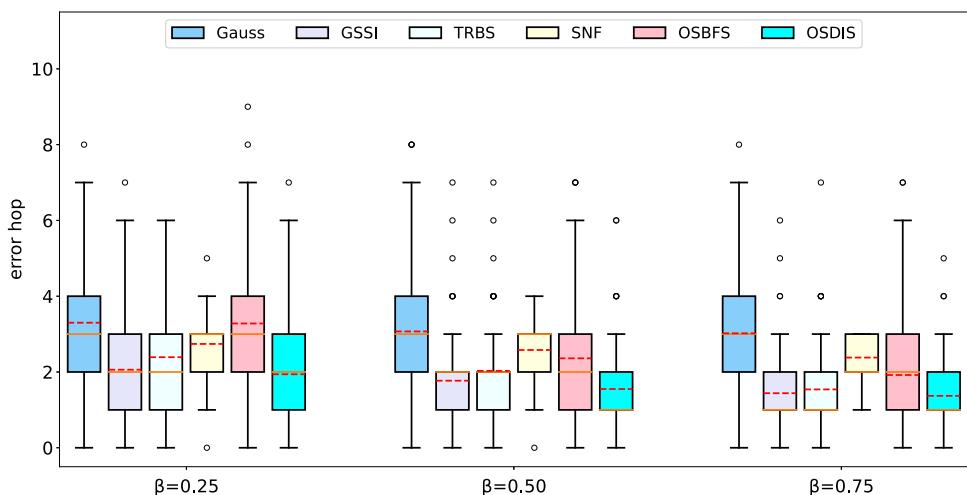
**Fig. 24** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on PDZBase network. The orange line and red line in each box denote the median and average error hop, respectively (color figure online)



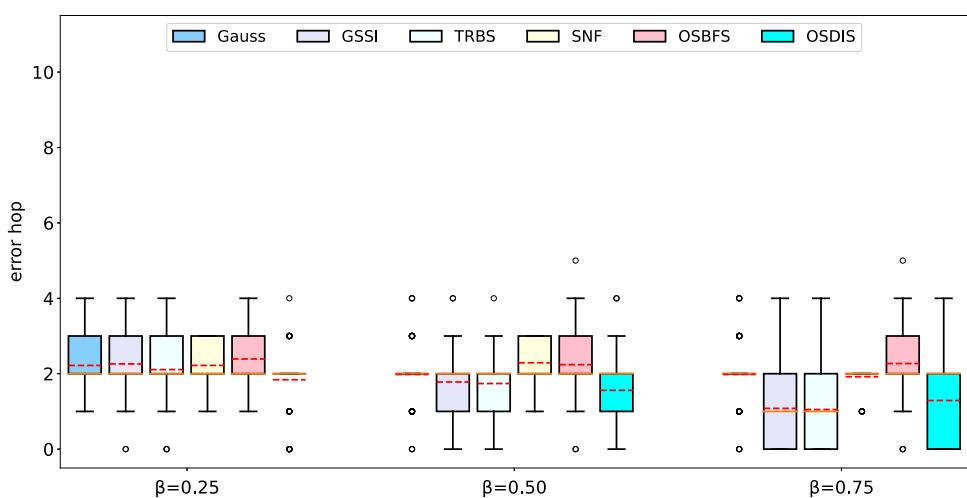
**Fig. 25** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on USAirlines network. The orange line and red line in each box denote the median and average error hop, respectively (color figure online)



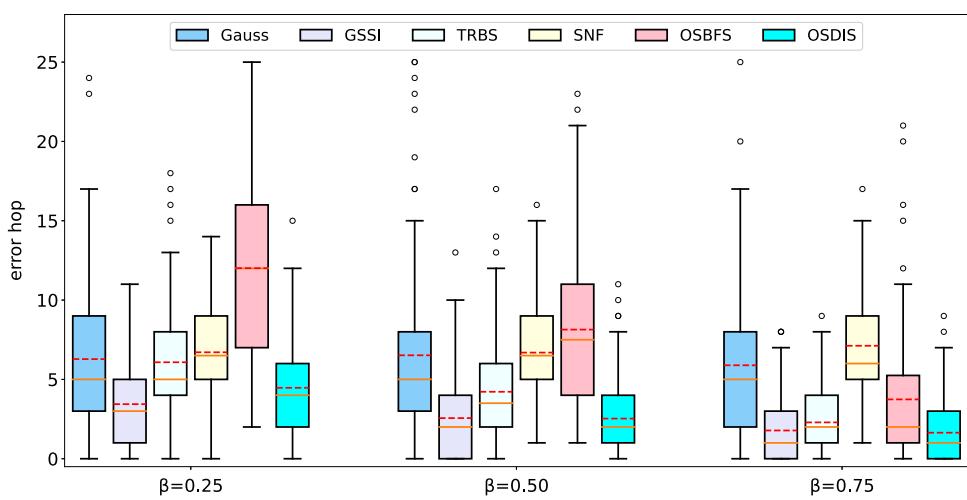
**Fig. 26** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on NetScience network. The orange line and red line in each box denote the median and average error hop, respectively (color figure online)



**Fig. 27** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on C elegans network. The orange line and red line in each box denote the median and average error hop, respectively (color figure online)



**Fig. 28** The IQR (box-plot) of error hop of Gauss, GSSI, TRBS, SNF, OSBFS and OSDIS algorithms on Euroroads network. The orange line and red line in each box denote the median and average error hop, respectively (color figure online)



## References

- Altarelli F, Braunstein A, Dall'Asta L, Lage-Castellanos A, Zecchina R (2014) Bayesian inference of epidemics on networks via belief propagation. *Phys Rev Lett* 112(11):118701
- Antulov-Fantulin N, Lančić A, Šmuc T, Štefančić H, Šikić M (2015) Identification of patient zero in static and temporal networks: robustness and limitations. *Phys Rev Lett* 114(24):248701
- Barabasi AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286(5439):509–512
- Brockmann D, Helbing D (2013) The hidden geometry of complex, network-driven contagion phenomena. *Science* 342(6164):1337–1342
- Cai K, Hong X, Lui JCS (2018) Information spreading forensics via sequential dependent snapshots. *IEEE/ACM Trans Netw* 26(1):478–491
- Caputo JG, Hamdi A, Knippel A (2019) Inverse source problem in a forced network. *Inverse Probl* 35(5):055006
- Chai Y, Wang Y, Zhu L (2021) Information sources estimation in time-varying networks. *IEEE Trans Inf Forens Secur* 99:2621–2636
- Chang B, Chen E, Zhu F, Liu Q, Xu T, Wang Z (2020) Maximum a posteriori estimation for information source detection. *IEEE Trans Syst Man Cybern Syst* 50(6):2242–2256
- Feizi S, Médard M, Quon G, Kellis M, Duffy K (2019) Network infusion to infer information sources in networks. *IEEE Trans Netw Sci Eng* 6(3):402–417
- Fu L, Shen Z, Wang WX, Fan Y, Di Z (2016) Multi-source localization on complex networks with limited observers. *EPL* 113(1):18006
- Horn AL, Friedrich H (2019) Locating the source of large-scale outbreaks of foodborne disease. *J R Soc Interface* 16(151):20180624
- Hosseini S, Azgomi MA (2016) A model for malware propagation in scale-free networks based on rumor spreading process. *Comput Netw* 108:97–107
- Hu Z, Wang L, Tang C (2019) Locating the source node of diffusion process in cyber-physical networks via minimum observers. *Chaos Interdiscip J Nonlinear Sci* 29(6):063117
- Jiang J, Sheng W, Shui Y, Yang X, Zhou W (2017) Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Commun Surv Tutor* 19(1):465–481
- Jiang J, Wen S, Yu S, Xiang Y, Zhou W (2018) Rumor source identification in social networks with time-varying topology. *IEEE Trans Dependable Secure Comput* 15(1):166–179
- Jiang J, Wen S, Yu S, Xiang Y, Zhou W (2015) K-center: An approach on the multi-source identification of information diffusion. *IEEE Trans Inf Forensics Secur* 10(12):2616–2626
- Kendall MG (1938) A new measure of rank correlation. *Biometrika* 30(1/2):81–93
- Kunegis J (2013) Konect: The koblenz network collection. In: Proceedings of the 22nd International Conference on World Wide Web, New York, NY, USA, pp 1343–1350
- Li J, J M, Bertuzzo E, Kolaczyk E D, (2021) Sensor-based localization of epidemic sources on human mobility networks. *PLoS Comput Biol* 17(1):e1008545
- Lokhov AY, Mézard M, Ohta H, Zdeborová L (2014) Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Phys Rev E* 90(1):012801
- Luo W, Tay WP, Leng M (2013) Identifying infection sources and regions in large networks. *IEEE Trans Signal Process* 61(11):2850–2865. <https://doi.org/10.1109/TSP.2013.2256902>
- Luo W, Tay WP, Leng M (2014) How to identify an infection source with limited observations. *IEEE J Sel Top Signal Process* 8(4):586–597
- Manitz J, Harbering J, Schmidt M, Kneib T, Schöbel A (2017) Source estimation for propagation processes on complex networks with an application to delays in public transportation systems. *J R Stat Soc Series C Appl Stat* 66(3):521–536
- Newman MEJ (2002) Assortative mixing in networks. *Phys Rev Lett* 89(20):208701
- Paluch R, Lu X, Suchecki K, Szymański BK, Holyst JA (2018) Fast and accurate detection of spread source in large complex networks. *Sci Rep* 8(1):2508
- Paluch R, Gajewski Łukasz G, Holyst JA, Szymanski BK (2020) Optimizing sensors placement in complex networks for localization of hidden signal source: A review. *Future Gen Comput Syst* 112:1070–1092
- Pinto PC, Patrick T, Martin V (2012) Locating the source of diffusion in large-scale networks. *Phys Rev Lett* 109(6):068702
- Prakash BA, Vreeken J, Faloutsos C (2014) Efficiently spotting the starting points of an epidemic in a large graph. *Knowl Inf Syst* 38(1):35–59
- Rossi RA, Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, pp 4292–4293, <https://ojs.aaai.org/index.php/AAAI/article/view/9277>
- Rui X, Wunsch D (2005) Survey of clustering algorithms. *IEEE Trans Neural Netw* 16(3):645–678
- Shah D, Zaman T (2011) Rumors in a network: Who's the culprit? *IEEE Trans Inf Theory* 57(8):5163–5181
- Shen Z, Cao S, Wang WX, Di Z, Stanley HE (2016) Locating the source of diffusion in complex networks by time-reversal backward spreading. *Phys Rev E* 93(3):032301
- Tang W, Ji F, Tay WP (2018) Estimating infection sources in networks using partial timestamps. *IEEE Trans Inf Forensics Secur* 13(12):3035–3049
- Tian H, Liu Y, Li Y, Wu CH, Chen B, Kraemer MUG, Li B, Cai J, Xu B, Yang Q, Wang B, Yang P, Cui Y, Song Y, Zheng P, Wang Q, Bjornstad ON, Yang R, Grenfell BT, Pybus OG, Dye C (2020) An investigation of transmission control measures during the first 50 days of the covid-19 epidemic in china. *Science* 368(6491):638–642
- Wang H (2019) An universal algorithm for source location in complex networks. *Phys A Stat Mech Appl* 514:620–630
- Wang H, Sun K (2020) Locating source of heterogeneous propagation model by universal algorithm. *EPL Europhys Lett* 131(4):48001
- Wang HJ, Zhang FF, Sun KJ (2021) An algorithm for locating propagation source in complex networks. *Phys Lett A* 393:127184
- Wang Y, Wen S, Xiang Y, Zhou W (2014) Modeling the propagation of worms in networks: A survey. *IEEE Commun Surv Tutor* 16(2):942–960
- Wang Z, Dong W, Zhang W, Tan CW (2014) Rumor source detection with multiple observations: Fundamental limits and algorithms. *SIGMETRICS Perform Eval Rev* 42(1):1–13
- Wang Z, Dong W, Zhang W, Tan CW (2015) Rooting our rumor sources in online social networks: The value of diversity from multiple observations. *IEEE J Sel Top Signal Process* 9(4):663–677
- Wang Z, Hou D, Gao C, Huang J, Xuan Q (2022) A rapid source localization method in the early stage of large-scale network propagation. In: Proceedings of the ACM Web Conference 2022, New York, NY, USA, pp 1372–1380
- Watts DJ, Strogatz SH (1998) Collective dynamics of ‘small-world’ networks. *Nature* 393(6684):440
- Xu S, Teng C, Zhou Y, Peng J, Zhang ZK (2019) Identifying the diffusion source in complex networks with limited observers. *Phys A Stat Mech Appl* 527:121267. <https://doi.org/10.1016/j.physa.2019.121267>
- Yang F, Zhang R, Yao Y, Yuan Y (2016) Locating the propagation source on complex networks with propagation centrality algorithm. *Knowl Based Syst* 100:112–123
- Yang F, Yang S, Peng Y, Yao Y, Wang Z, Li H, Liu J, Zhang R, Li C (2020) Locating the propagation source in complex networks with

- a direction-induced search based gaussian estimator. *Knowl Based Syst* 195:105674
- Yu F, Xia X, Li W, Tao J, Ma L, Cai Z (2017) Critical node identification for complex network based on a novel minimum connected dominating set. *Soft Comput* 21(19):5621–5629
- Zhang X, Wu J, Zhao P, Su X, Choi D (2018) Epidemic spreading on a complex network with partial immunization. *Soft Comput* 22(14):4525–4533
- Zhu K, Ying L (2014) A robust information source estimator with sparse observations. *Comput Social Netw* 1(1):1–21
- Zhu K, Ying L (2016) Information source detection in the sir model: A sample-path-based approach. *IEEE/ACM Trans Netw* 24(1):408–421

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.