Source detection in simplicial complexes based on dynamic message passing algorithm

Ling Kang(康玲), Ting Jiang(江婷), Hai-Feng Zhang(张海峰), Huan Wang(王欢)

Citation: Chin. Phys. B. 35, 108901 (2026). DOI: 10.1088/1674-1056/ae4e89 | CSTR: 32038.14.CPB.ae4e89
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Competitive higher-order network evolution model

Jin-Li Guo(郭进利)
Chin. Phys. B. 35, 088901 (2026). DOI: 10.1088/1674-1056/ae42b6 | CSTR: 32038.14.CPB.ae42b6

SLGC: Identifying influential nodes in complex networks from the perspectives of self-centrality, local centrality, and global centrality

Da Ai(艾达), Xin-Long Liu(刘鑫龙), Wen-Zhe Kang(康文哲), Lin-Na Li(李琳娜), Shao-Qing Lü(吕少卿), Ying Liu(刘颖)
Chin. Phys. B. 32, 118902 (2023). DOI: 10.1088/1674-1056/aceee8

Identifying multiple influential spreaders in complex networks based on spectral graph theory

Dong-Xu Cui(崔东旭), Jia-Lin He(何嘉林), Zi-Fei Xiao(肖子飞), Wei-Ping Ren(任卫平)
Chin. Phys. B. 32, 098904 (2023). DOI: 10.1088/1674-1056/acac0b

An extended improved global structure model for influential node identification in complex networks

Jing-Cheng Zhu(王伦文), Lun-Wen Wang(王伦文)
Chin. Phys. B. 31, 068904 (2022). DOI: 10.1088/1674-1056/ac380d

A novel method for identifying influential nodes in complex networks based on gravity model

Yuan Jiang(蒋沅), Song-Qing Yang(杨松青), Yu-Wei Yan(严玉为), Tian-Chi Tong(童天驰), Ji-Yang Dai(代冀阳)
Chin. Phys. B. 31, 058903 (2022). DOI: 10.1088/1674-1056/ac4226

Source detection in simplicial complexes based on dynamic message passing algorithm

Ling Kang(康玲)[1], Ting Jiang(江婷)[2], Hai-Feng Zhang(张海峰)[2], and Huan Wang(王欢)[3,†]

[1] *School of Medical Informatics Engineering, Anhui University of Chinese Medicine, Hefei* 230012, *China*
[2] *School of Mathematical Science, Anhui University, Hefei* 230601, *China*
[3] *School of Big Data and Statistics, Anhui University, Hefei* 230601, *China*

(Received 3 January 2026; revised manuscript received 28 February 2026; accepted manuscript online 7 March 2026)

The source detection problem, which aims to identify the seeds of false information propagating on social networks, has attracted extensive attention in the fields of network science and security. Previous information propagation models mainly focus on simple graphs (involving only pairwise interactions) and ignore the social reinforcement effect during the information propagation process. To address this, we build up a susceptible–infected–recovered (SIR) propagation model on simplicial complexes in this paper, which features the social reinforcement effect generated from higher-order interactions. Based on this model, we further propose a simplicial complex dynamic message passing (SCDMP) algorithm, and then develop an energy-based metric to infer the propagation source within simplicial complex networks. Experiments on real-world and synthetic networks demonstrate that, compared with baseline algorithms, the SCDMP algorithm exhibits significant advantages in detection accuracy, stability, and robustness.

Keywords: source detection problem, simplicial complex, dynamic message passing, SIR model

PACS: 89.75.Fb

DOI: 10.1088/1674-1056/ae4e89 **CSTR:** 32038.14.CPB.ae4e89

1. Introduction

The widespread propagation of misinformation on social media may lead to a series of societal risks, including public health threats,[1] personal information leaks,[2] and uncontrolled public discourse.[3] To address such issues, researchers have increasingly developed source detection methods, aiming to infer the origins of misinformation based on observed information propagation processes. Source detection aims to trace the origin of propagation by analyzing data such as node states, propagation paths, or temporal information. This approach holds significant theoretical and practical value, which is applicable not only to the timely control of rumors and misinformation in social networks but also to locating propagation sources in infectious disease prevention and control. Precise identification of information sources provides critical evidence for public opinion guidance and risk response, enabling more efficient and proactive prevention and control within complex networks. Furthermore, continuous optimization of source detection techniques offers new insights and methodological support for interdisciplinary research.[4]

In network topology-based source detection methods, researchers generally recognize that propagation sources typically exhibit high centrality within propagation subgraphs. Thus, identifying nodes with high centralities potentially helps infer source nodes.[5] For instance, Comin *et al.*[6] introduced metrics such as degree centrality, betweenness centrality, closeness centrality, and eigenvector centrality to locate propagation sources, spurring multiple subsequent extension studies.[7,8] To measure the centrality of nodes as sources within propagation subgraphs, Shah *et al.*[9,10] proposed a source detection algorithm based on the SI model, termed "rumor centrality". Its core principle is that if a node is an information source, the number of possible propagation sequences originating from it should be proportional to the actual infection scale. Building upon this, Luo *et al.*[11] extended Shah's approach by introducing Jordan centrality to calculate node centrality within propagation subgraphs. They identified Jordan nodes as the most probable propagation sources, demonstrating superior detection accuracy compared to the rumor centrality. Meanwhile, Zhu *et al.*[12] proposed a reverse infection algorithm using approximation methods, which identifies sources closer to actual origins than proximity-based heuristics. Wang *et al.*[13] introduced the label propagation source identification (LPSI) algorithm, identifying nodes with the highest local label values as sources. Deng *et al.*[14] developed a backward propagation centrality algorithm to identify all information sources triggering propagation at different start times.

Beyond subgraph centrality analysis, propagation-based methods also estimate sources. In particular, Altarelli *et al.*[15] developed a trust propagation algorithm for SIR models, detecting source nodes by deriving posterior probability distributions of node state evolution. However, this approach strug-

†Corresponding author. E-mail: hwang1106@163.com

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

gles to converge when faced with looped networks. Lokhov *et al.*[16] developed the dynamic message passing (DMP) algorithm. Compared to traditional Bayesian propagation (BP), this approach more directly derives system state equations, enabling efficient estimation of marginal probabilities in tree-like and sparse cyclic graphs. It is also extensible to Bayesian networks[17] and optimization problems. This method calculates the probability of each node's state at the observation time, assuming the propagation source.[18] Subsequently, Jiang *et al.*[19] proposed a symbolic-dynamic message passing (S-DMP) algorithm for source localization in symbolic networks. Shi *et al.*[20] formulated the source identification problem as a maximum likelihood (ML) estimator and introduced a Monte Carlo simulation (MCS)-based statistical detection method to estimate source locations and initial propagation times. Ma *et al.*[21] transformed propagation probabilities into effective distances for observer selection, proposing a backpropagation-based source localization method for symbolic networks. Other approaches include evidence-theoretic algorithms,[22] probabilistic detection methods,[23] and maximum entropy sampling.[24] The continuous emergence of propagation source detection methods provides crucial support for identifying erroneous information sources on graphs.[25]

Recently, deep learning methods have been widely applied to source detection problems due to their ability to implicitly learn propagation patterns. For example, building upon graph neural networks,[26] Ran *et al.*[27] proposed a Multi-channel Graph ATtention network (MGAT) hybrid architecture that significantly enhances performance by integrating network topology with node features. Čutura *et al.*[28] proposed the conditional variational autoencoder (DDmix), a probabilistic method based on constrained learning, which can be trained on past propagation patterns to capture latent propagation dynamics. Additionally, specialized solutions tailored for specific scenarios have emerged, such as the multi-source fusion spatio–temporal graph convolutional network (MSF-STGCN) framework[29] designed for source detection in dynamic networks. These deep learning-based methods sufficiently improved the detection accuracy, but their practical applications are limited due to large-scale training and high computational complexity.

Although source detection has been extensively studied over the past decade, most existing methods are limited to traditional networks that consider only pairwise interactions between different nodes. Many realistic systems exhibit group-level interactions beyond simple pairwise connections.[30] Higher-order structures significantly influence propagation dynamics, prompting us to develop source detection methods for infectious disease processes on advanced structures like pure complexes and hypergraphs. In recent years, there has been growing attention focused on source detection with hypergraphs. For instance, Yu *et al.*[31] proposed a DMP algorithm based on factor graphs, achieving mathematical equivalence between the hypergraph and its factor graph. He *et al.*[32] introduced a heterogeneous hypergraph attention network (HHAN) model, which explains multiple propagation pathways and patterns within heterogeneous networks. Liang *et al.*[33] also tackled the vital node identification problem in hypergraphs, proposing the LGK method that leverages higher-order structures. However, hypergraph models face limitations when studying social contagion processes such as opinion propagation and behavioral imitation. For instance, Battiston *et al.*[34] provided a systematic review of higher-order networks, noting that hypergraphs may exhibit structural redundancy when capturing the mechanism of "simultaneous multi-source exposure." Iacopini *et al.*[35] pointed out that the non-closed structure of hypergraphs makes it difficult to naturally characterize such social reinforcement mechanisms that rely on higher-order interactions. Fan *et al.*[36] emphasized that complex contagion fundamentally depends on a reinforcement mechanism requiring multiple simultaneous sources of exposure.

As noted by Centola and Macy,[37] complex contagions typically require individuals to encounter two or more sources of exposure before propagation occurs, and thus fundamentally rely on a reinforcement mechanism in which multiple identical stimuli are received simultaneously. Simplicial complexes provide a more suitable framework for capturing this mechanism. In a simplicial complex, a K-simplex consists of $K+1$ nodes, and every subset of these nodes is also required to form a simplex. By contrast, hypergraph models differ in a fundamental way. When four researchers form a single hyperedge, the hyperedge merely indicates that these four authors have collaborated, without requiring that all subsets of them have also collaborated. Although existing hypergraph-based methods (such as DMP based on factor graphs or heterogeneous attention networks) can address higher-order interactions to some extent, their underlying structure still struggles to strictly characterize the critical dependency of "simultaneous multi-source exposure." Therefore, owing to its closure property, a simplicial complex offers a more faithful representation of the reinforcement mechanisms underlying complex contagions, as it explicitly encodes multiple simultaneous sources of exposure while avoiding the redundant or spurious connections that may arise in hypergraph representations.

While simplicial complexes offer a principled way to capture the social reinforcement effects that fundamentally shape information propagation, existing studies on source detection over simplicial complexes remain scarce. This gap is significant, as traditional graph-based methods fail to account for higher-order interactions that are essential for modeling complex contagions. Meanwhile, efforts to identify influential

spreaders in simple graphs have begun to incorporate community structure and entropy measures, such as the DECS algorithm by Su *et al.*,[38] which uses the Walktrap algorithm and density entropy. To bridge this gap, we extend the dynamic message passing paradigm from pairwise graphs to the higher-order structure of simplicial complexes. Based on this idea, we propose a SCDMP algorithm, which enables effective detection with infection sources by explicitly leveraging higher-order propagation patterns. Through extensive experiments on both synthetic and real-world networks, we show that SCDMP achieves substantial performance gains over existing baselines, highlighting the importance of incorporating higher-order topology in source localization. These results demonstrate that the proposed SCDMP algorithm not only fills a critical methodological gap but also establishes a new framework for source detection in higher-order spreading processes.

The remainder of this paper is organized as follows. Section 2 introduces the SIR propagation model on simplex complexes. Section 3 details the SCDMP algorithm. Section 4 presents baseline methods and corresponding numerical experiments. Finally, Section 5 concludes the paper. Table 1 lists some key symbols and definitions used in this paper.

Table 1. Notations and definitions.

Notation	Definition
V	Set of vertices
C	Set of simplices
β_n	Infection probability for an n-simplex
O_t	Time snapshot
$P_S^i(t)$	Probability that node i is in state S at time t
$P_I^i(t)$	Probability that node i is in state I at time t
$P_R^i(t)$	Probability that node i is in state R at time t
$\theta^{j\to i}(t)$	Blocking probability from j to i via a 1-simplex (edge) by time t
$\theta^{k,l\to i}(t)$	Blocking probability from k and l to i via a 2-simplex until time t
$\theta^{a,b,c\to i}(t)$	Blocking probability from a, b, and c to i via a 3-simplex until time t
$\Gamma(i)$	Neighbor set of node i
$\Delta(i)$	Other two nodes in the 2-simplex containing node i
$\Omega(i)$	Other three nodes in the 3-simplex containing node i
μ	Recovery probability
$\phi^{j\to i}(t)$	Probability that node j is infected at time t but does not infect i through an edge
$\phi^{k,l\to i}(t)$	Probability that nodes k and l are infected at time t but do not infect i through a 2-simplex
$\phi^{a,b,c\to i}(t)$	Probability that nodes a, b, and c are infected at time t but do not infect i through a 3-simplex
$d_G(i,v)$	Shortest distance between i and v in graph G
$\langle k_1\rangle$	Average degree of nodes in 1-simplex
N	Number of nodes in the network
k_2	Average degree of nodes in 2-simplex
β_1^c	Epidemic network propagation threshold

2. Definitions

This section defines simplex complexes, introduces propagation models on simplex complexes, and addresses the source detection problem on simplex complexes.

2.1. Definition of simplex complexes

A simplex complex is defined as $M=(V,C)$, where V denotes the set of vertices and C denotes the set of simplices. A kth-order simplex $\sigma\subset C$ represents a higher-order interaction involving $k+1$ vertices. As shown in Fig. 1(a), within a simplex complex, 0th-order simplices represent individual vertices, 1st-order simplices represent edges, and second-order simplices represent "full" triangular faces, etc. A simplex complex satisfies the closure condition: if a simplex $\sigma\subset C$, then all sub-simplices $\varepsilon\subset\sigma$ must be included in C. For instance, if the second-order simplex $\omega=\{v_0,v_1,v_2\}\subset C$, then $\{v_0\}$, $\{v_1\}$, $\{v_2\}$, $\{v_0,v_1\}$, $\{v_0,v_2\}$, and $\{v_1,v_2\}$ are also in C.

2.2. The SIR model on simplicial complexes

To characterize social reinforcement mechanisms during propagation, this paper defines an SIR propagation model on simplex complexes. In this model, each node exists in one of three states: susceptible (S), infected (I), or recovered (R). Infected nodes transmit infection to their susceptible neighbors. The process proceeds as follows:

(1) At $t=0$, activate nodes to obtain a seed set and a set of inactivated nodes;

(2) $t=1$: Set the edge infection probability to β_1, the 2-simplex infection probability to β_2, …, and the n-simplex infection probability to β_n. For each inactive node v, infect it via the following possible pathways:

a) If v is within a 1-simplex (u,v) and u is an infected node, u infects v with probability β_1. As shown in Figs. 1(b)–1(g), red edges represent S-state individuals being infected via 1-simplex with probability β_1.

b) If v is in a 2-simplex (u,v,l) and both u and l are infected nodes, v can be infected with two β_1 probabilities plus an additional 2-simplex probability β_2. The red "filled" triangular faces in Fig. 1(f) represent S-state individuals being infected via a 2-simplex with probability β_2. Similarly, for a node v in an n-simplex, if all other nodes in the n-simplex except v are infected, then v is infected with an n-simplex infection probability β_n.

Each node infected in the previous time step transitions to the recovered state with recovery probability μ, as shown in Fig. 1(h).

(3) Repeat step (2) until time step T is reached.

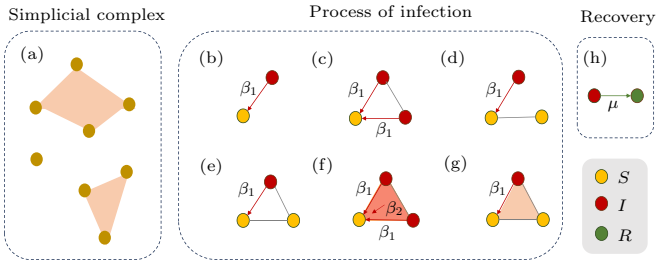


Fig. 1. Simple complex SIR propagation model.

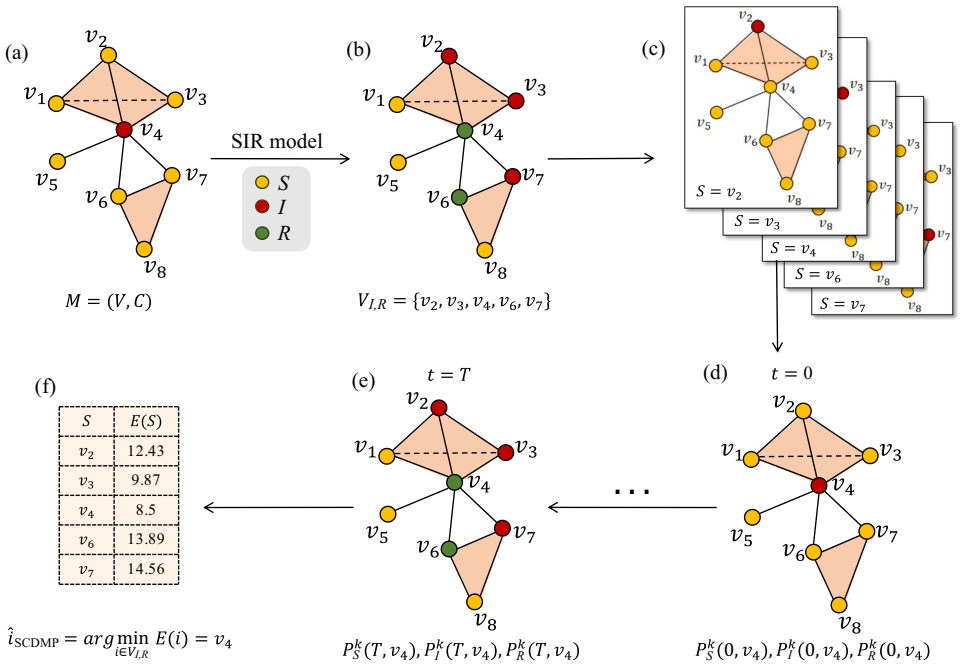
2.3. Source detection problem on simplex complexes

For a given simplex complex $M = (V,C)$, based on the SIR model, an infection signal propagates from a single node $i \in V$. After time t, we can only observe the state of nodes at time t (the time snapshot O_t). Different source nodes generate distinct state snapshots at time t. The goal of source detection is to infer which node i is most likely the source using the snapshot state information O_t and the network topology information M. Mathematically, it can be expressed as

$$s^* = \arg\max_{i \in V} P(O_t \mid i, M). \tag{1}$$

3. Message passing algorithm based on simplicial complexes

After the simplicial complex network undergoes the SIR propagation model, all nodes in the network are in the S, I, or R state, as shown in Fig. 2(b). In this paper, we generalize to simplicial complex networks with up to 3-simplex. For node i, we define $P_S^i(t)$, $P_I^i(t)$ and $P_R^i(t)$ to represent the probabilities that node i is in each state at time t, respectively.



S	$E(S)$
v_2	12.43
v_3	9.87
v_4	8.5
v_6	13.89
v_7	14.56

Fig. 2. Framework diagram of the SCDMP algorithm. (a) A simplicial complex network with node v_4 as the source node. (b) The state of each node at the final time T obtained by propagation according to the simplicial complex SIR propagation model. (c)–(e) According to each candidate source node (infected node), the probabilities of each node being in the three states at the final time T are obtained by iterating from the initial time through the simplicial complex dynamic message passing algorithm. (f) The most likely source node is obtained according to the objective equation.

In a simplicial complex, if node i is in the S state at time t, it means that, on the one hand, the node is in the S state at the initial time $t = 0$ (denoted as $P_S^i(0)$), and on the other hand, all nodes connected to it and all simplices it belongs to have not transmitted infection information to it until time t. Specifically, we define $\theta^{j\to i}(t)$ as the blocking probability that the propagation information is not propagated from node j to node i through a 1-simplex (edge) until time t, $\theta^{k,l\to i}(t)$ represents the blocking probability that the propagation information is not propagated from nodes k and l to node i through a 2-simplex until time t, and $\theta^{a,b,c\to i}(t)$ denotes the blocking probability that the propagation information is not propagated from nodes a, b, and c to node i through a 3-simplex until time t. It is noted that a node cannot self-propagate. Therefore, we

obtain the probability that node i is in the S state at time t:

$$P_S^i(t) = P_S^i(0)\prod_{j\in\Gamma(i)}\theta^{j\to i}(t)\prod_{k,l\in\Delta(i)}\theta^{k,l\to i}(t) \times\prod_{a,b,c\in\Omega(i)}\theta^{a,b,c\to i}(t), \tag{2}$$

where $\Gamma(i)$ is the neighbor set of node i, $\Delta(i)$ represents the two nodes (excluding node i) in the set of 2-simplex that node i belongs to, and $\Omega(i)$ denotes the three nodes (excluding node i) in the set of 3-simplex that node i belongs to. The first product term represents that node i is not infected by an edge, the second product term represents that node i is not infected by a 2-simplex, and the third product term represents that node i is not infected by a 3-simplex.

Node i is in state R at time t if it was already in state R at $t-1$, or if it was in state I at $t-1$ and recovers with probability μ. Once recovered, it will remain in the recovered state permanently, which is expressed as

$$P_R^i(t) = P_R^i(t-1) + \mu P_I^i(t-1). \tag{3}$$

Then, the probability that node i is in the infected state at time t is

$$P_I^i(t) = 1 - P_S^i(t) - P_R^i(t). \tag{4}$$

To calculate the three blocking probabilities in Eq. (2), we introduce three additional probabilities. $\phi^{j\to i}(t)$: the probability that node j is in the infected state at time t but does not infect node i through an edge; $\phi^{k,l\to i}(t)$: the probability that nodes k and l are in the infected state at time t but do not infect node i through a 2-simplex; $\phi^{a,b,c\to i}(t)$: the probability that nodes a, b, and c are in the infected state at time t but do not infect node i through a 3-simplex.

Therefore, the updated formulas for the blocking probabilities of edges, 2-simplex, and 3-simplex are

$$\theta^{j\to i}(t+1) - \theta^{j\to i}(t) = -\beta_1\phi^{j\to i}(t), \tag{5}$$

$$\theta^{k,l\to i}(t+1) - \theta^{k,l\to i}(t) = -\beta_2\phi^{k,l\to i}(t), \tag{6}$$

$$\theta^{a,b,c\to i}(t+1) - \theta^{a,b,c\to i}(t) = -\beta_3\phi^{a,b,c\to i}(t). \tag{7}$$

There are three cases for the change of $\phi^{j\to i}(t)$: 1) Node j infects node i through an edge; 2) Node j transitions to the recovered state; 3) Node j changes from state S to state I from time $t-1$ to time t. The update formula for $\phi^{j\to i}(t)$ is obtained as

$$\phi^{j\to i}(t) = (1-\beta_1)(1-\mu)\phi^{j\to i}(t-1) + P_S^{j\to i}(t-1) - P_S^{j\to i}(t), \tag{8}$$

where the term $(1-\beta_1)(1-\mu)\phi^{j\to i}(t-1)$ represents that j neither recovers nor infects node i through an edge from time $t-1$ to time t, and $P_S^{j\to i}(t-1) - P_S^{j\to i}(t)$ represents that j is infected from time $t-1$ to time t, which needs to be added to reflect this new infection non-propagation state.

For the change of $\phi^{k,l\to i}(t)$: 1) k and l infect i through a simplex; 2) Both k and l recover; 3) One of k and l recovers while the other does not; 4) k is always in state I, and l changes from state S to state I from time $t-1$ to time t; 5) l is always in state I, and k changes from state S to state I from time $t-1$ to time t; 6) Both k and l were in state S before and change to state I at this time. Among them, the term $\beta_2\mu^2 + 2\beta_2\mu(1-\mu)$ acts as a compensation term, accounting for the scenarios where k and l infect i at the same time and both transition to the recovered state, or infect i at the same time with one of them transitioning to the recovered state.

$$\begin{aligned}\phi^{k,l\to i}(t) = {} & \big(1-\beta_2-\mu^2-2\mu(1-\mu)+\beta_2\mu^2+2\beta_2\mu(1-\mu)\big)\phi^{k,l\to i}(t-1) + (1-\mu)P_I^{k\to i}(t-1)(P_S^{l\to i}(t-1) - P_S^{l\to i}(t)) \\ & + (P_S^{k\to i}(t-1) - P_S^{k\to i}(t))P_I^{l\to i}(t-1)(1-\mu) + (P_S^{l\to i}(t-1) - P_S^{l\to i}(t))(P_S^{k\to i}(t-1) - P_S^{k\to i}(t)).\end{aligned} \tag{9}$$

Simplifying Eq. (9), we have

$$\begin{aligned}\phi^{k,l\to i}(t) = {} & (1-\beta_2)(1-\mu)^2\phi^{k,l\to i}(t-1) + (1-\mu)P_I^{k\to i}(t-1)(P_S^{l\to i}(t-1) - P_S^{l\to i}(t)) \\ & + (P_S^{k\to i}(t-1) - P_S^{k\to i}(t))P_I^{l\to i}(t-1)(1-\mu) + (P_S^{l\to i}(t-1) - P_S^{l\to i}(t))(P_S^{k\to i}(t-1) - P_S^{k\to i}(t)).\end{aligned} \tag{10}$$

Similarly, we can derive the update formula for $\phi^{a,b,c\to i}(t)$:

$$\begin{aligned}\phi^{a,b,c\to i}(t) = {} & (1-\beta_3)(1-\mu)^3\phi^{a,b,c\to i}(t-1) + (1-\mu)^2P_I^{a\to i}(t-1)P_I^{b\to i}(t-1)(P_S^{c\to i}(t-1) - P_S^{c\to i}(t)) \\ & + (1-\mu)^2P_I^{a\to i}(t-1)P_I^{c\to i}(t-1)(P_S^{b\to i}(t-1) - P_S^{b\to i}(t)) \\ & + (1-\mu)^2P_I^{b\to i}(t-1)P_I^{c\to i}(t-1)(P_S^{a\to i}(t-1) - P_S^{a\to i}(t)) \\ & + (1-\mu)P_I^{a\to i}(t-1)(P_S^{b\to i}(t-1) - P_S^{b\to i}(t))(P_S^{c\to i}(t-1) - P_S^{c\to i}(t)) \\ & + (1-\mu)P_I^{b\to i}(t-1)(P_S^{a\to i}(t-1) - P_S^{a\to i}(t))(P_S^{c\to i}(t-1) - P_S^{c\to i}(t)) \\ & + (1-\mu)P_I^{c\to i}(t-1)(P_S^{b\to i}(t-1) - P_S^{b\to i}(t))(P_S^{a\to i}(t-1) - P_S^{a\to i}(t)) \\ & + (P_S^{a\to i}(t-1) - P_S^{a\to i}(t))(P_S^{b\to i}(t-1) - P_S^{b\to i}(t))(P_S^{c\to i}(t-1) - P_S^{c\to i}(t)),\end{aligned} \tag{11}$$

wherein the first term on the right-hand side of the equation represents that nodes a, b, and c maintain their state at time $t-1$ (i.e., none of them recovers and they do not infect node i through a 3-simplex) until time t; the second, third, and fourth terms on the right-hand side of the equation indicate that among nodes a, b, and c, two were in state I at the previous time and remain in that state until time t, while the other one changes from state S to state I from time $t-1$ to time t; the fifth, sixth, and seventh terms on the right-hand side of the equation denote that among nodes a, b, and c, one was in state I at the previous time and remains in that state until time t, while the other two change from state S to state I from time $t-1$ to time t; the last term represents that all nodes a, b, and c change from state S to state I from time $t-1$ to time t.

Consider an auxiliary dynamic D_i: node i is always in state S, that is, the state of node i is not considered. Under event D_i, i will not cause any infection, which enables us to assume that the neighboring branches of i are mutually independent. Then, under dynamic D_i, the probability that node j is in state S at time t is

$$P_S^{j\to i}(t) = P_S^j(0)\prod_{m\in\Gamma(j)\setminus i}\theta^{m\to j}(t)\prod_{x,y\in\Delta(j)\setminus i}\theta^{x,y\to j}(t) \times\prod_{a,b,c\in\Omega(j)\setminus i}\theta^{a,b,c\to j}(t). \tag{12}$$

Using the update formulas derived above, we iterate from the initial state

$$\theta^{i\to j}(0)=\theta^{k,l\to j}(0)=\theta^{a,b,c\to j}(0)=1, \tag{13}$$

$$\phi^{i\to j}(0)=P_I^i(0). \tag{14}$$

First, the initial values of the blocking probability θ in Eq. (13) are used to iteratively update the state probabilities of each node according to Eqs. (2)–(4). Then, the initial values of the probability θ in Eq. (14) are used to iteratively update the three types of blocking probabilities according to Eqs. (5)–(7). Finally, the probability θ is iteratively updated based on the state probabilities in Eqs. (8), (10)–(11). Through the above process, the probabilities of each node being in different states at different times can be obtained.

Given time t, using Bayes' rule, the probability that node i is the propagation source and the joint probability of the given snapshot observation state is

$$P(O_t\mid i,M)\cong\prod_{k\in O_t,q_k(t)=S}P_S^k(t,i)\prod_{l\in O_t,q_l(t)=I}P_I^l(t,i) \times\prod_{m\in O_t,q_m(t)=R}P_R^m(t,i), \tag{15}$$

O_t is the snapshot set, $q_k(t)=S$ indicates that node k is in state S at time snapshot t, and $P_S^k(t,i)$ represents the probability that k is in state S at time t when i is the propagation source. The probability calculated by this formula may be too small to compute. Therefore, we perform a logarithmic transformation on the result. To ensure the result is positive, we need to take the negative value. Finally, we obtain the expression defining this energy function:

$$\begin{aligned}E(i) &= -\log(P(O_t\mid i,M))\\ &= -\Bigg(\sum_{k\in O_t,q_k(t)=S}\log P_S^k(t,i)+\sum_{l\in O_t,q_l(t)=I}\log P_I^l(t,i)\\ &\quad+\sum_{m\in O_t,q_m(t)=R}\log P_R^m(t,i)\Bigg).\end{aligned} \tag{16}$$

Algorithm 1 SCDMP algorithm

Input: Adjacency matrix, 2-simplex adjacency tensor, 3-simplex adjacency tensor, $\beta_1,\beta_2,\beta_3,\mu$, Node state snapshot $\mathcal{S}_{\text{snap}}$, maximum time T
Output: Energy value E
1: Initialization: blocking probabilities, propagation message probabilities, node state probabilities, and edge state probabilities.
2: **for** $t=1$ to T
3: Set current time layer c and previous layer p.
4: **for** each node $u\in V$
5: Compute the product of blocking probabilities for edges, 2-simplex, and 3-simplex involving node u according to Eqs. (5)–(7).
6: Update the S, I, R state probabilities of node u according to Eqs. (2)–(4).
7: **for** each neighbor v of u
8: Update the edge state probability according to Eqs. (12)–(14).
9: Update the edge propagation message $\phi_{u\leftarrow v}$ according to Eq. (8).
10: **end for**
11: **for** each 2-simplex (u,v,w) containing u
12: Update the 2-simplex propagation message $\Phi_{u\leftarrow(v,w)}$ according to Eq. (10).
13: **end for**
14: **for** each 3-simplex (u,v,w,x) containing u
15: Update the 3-simplex propagation message $\Phi_{3;u\leftarrow(v,w,x)}$ using the 3D extension of Eq. (11).
16: **end for**
17: **end for**
18: **end for**
19: Compute the negative log-likelihood based on the snapshot $\mathcal{S}_{\text{snap}}$ and the node state probabilities at the final time to obtain the energy E.
20: **return** E

The lower the energy value, the more obvious the stability of the node state is reflected. Therefore, the current node is most likely to be the propagation source to achieve the current result. By sorting the nodes according to the energy values, different nodes can obtain corresponding sorting values. We define the node with the lowest sorting value as the propagation source node to be found, expressed by

$$\hat{i}_{\mathrm{SCDMP}} = \arg\min_{i \in V_{I,R}} E(i). \tag{17}$$

As shown in Figs. 2(d)–2(f), from the initial time to the snapshot time t, we sequentially perform calculations based on the above formulas, and finally obtain the state probabilities of each node at the snapshot time t, thereby solving the answer to the target Eq. (17). The algorithm of the SCDMP is given in Algorithm 1.

Specifically, the time complexity of the SCDMP algorithm primarily arises from the iterative message-passing process performed for each candidate source node. For a simplicial complex network with N nodes, average degree D, average number of 3-simplex per node K_3, and maximum propagation time T, the per-step computational complexity for each candidate source is $O(D + D^2 + K_3)$. When evaluating N candidate sources over T time steps, the total complexity is $O(N \cdot T \cdot (D + D^2 + K_3))$.

4. Experiments

4.1. Baseline methods

No prior work exists specifically addressing source detection in simplex complexes. Therefore, we employ five centrality-based methods to infer source nodes: Jordan centrality, eigenvector centrality, and label-based source identification algorithms — all designed for traditional graphs.[39] We treat the simplex complex M as a traditional graph G (i.e., considering only the 1-simplex structure). Furthermore, the higher-order closeness centrality and higher-order betweenness centrality are considered as the source detection algorithms for hypergraphs. To compare with them, we treat the simplex complex M as a hypergraph H (i.e., treating an n-simplex as $n+1$ hyperedges).

1. Jordan centrality (JC): In traditional graphs, a node's Jordan centrality is defined as the minimum distance to reach all other infected nodes and recovered nodes.[12] It is defined as

$$\mathrm{JC}(i, G) = \sum_{v \in V_{I,R}} d_G(i, v), \tag{18}$$

where $d_G(i, v)$ is the shortest distance between nodes i and v in the undirected graph G, where v is a node in either the I or R state.

2. Eigenvector centrality (EIG): This measure follows the principle that nodes connected to other important nodes in the network are more significant.[40,41] Mathematically, assuming x_j represents the importance of node j and A is the network's adjacency matrix, the importance of node i can be expressed as the sum of the importance values of all directly connected nodes j, where c is a proportionality constant. Therefore,

$$\mathrm{EIG}(i, G) = x_i = c \sum_{j \neq i}^{n} A_{ij} x_j. \tag{19}$$

3. Higher-order clustering coefficient (HCC): Its calculation is based on the subgraph formed by infected nodes, termed the "infected hypergraph" (H^I). The HCC value equals the reciprocal of the sum of the shortest path distances from the node to all other infected nodes within the H^I. The node set V^I of the H^I represents the infected nodes at time T. For each hyperedge in the original hypergraph H, a subset of nodes from V^I constitutes a hyperedge in the H^I. For example, if the original hyperedge $\sigma = \{v_1, v_2, v_3\}$ has only v_1 and v_2 infected at time T, then $\{v_1, v_2\}$ forms a hyperedge in the H^I. Isolated nodes are excluded from the H^I, and duplicate hyperedges are counted repeatedly.

Let n denote the number of nodes in the infected hypergraph H^I. To define the HCC, we first introduce the s-distance concept from distance calculation methods in hypergraphs.[42] Two hyperedges are s-adjacent if they share at least s nodes. Each pair of adjacent hyperedges is s-adjacent. The shortest s-distance $d_s^{\sigma}(p, q)$ between two hyperedges p and q is defined as the length of the shortest s-path connecting them. If no such s-path exists, their s-distance is defined as ∞. Based on this definition of hyperedge distance, we define the s-distance between nodes. Let σ_i denote the set of hyperedges containing node v_i. We define the distance between nodes v_i and v_j, denoted as $d_s^v(i, j)$, by

$$d_s^v(i, j) = \begin{cases} 1, & \text{nodes } i \text{ and } j \text{ are in the same hyperedge,} \\ \min_{p \in \sigma_i, q \in \sigma_j} d_s^{\sigma}(p, q) + 1, & \text{others.} \end{cases} \tag{20}$$

Ultimately, the higher-order centrality of node v_i, denoted as $\mathrm{HCC}(i)$, is given by the following equation:

$$\mathrm{HCC}(i) = \sum_{s=1}^{s_{\max}} \frac{1}{s^2} \sum_{j \neq i}^{n} \frac{1}{d_s^v(i, j)}, \tag{21}$$

where $s_{\max}$ is a parameter representing the maximum s-distance considered, while $1/s^2$ is a parameter controlling the contribution of different s-distances to HCC(i). This paper sets $s_{\max} = 3$.

4. Higher-order betweenness centrality (HBC): This metric extends the traditional network concept of betweenness centrality to the hypergraph. Specifically, we identify all s-paths within the infected hypergraph H^I. The higher-order betweenness centrality of node v_i is denoted as HBC(i), defined as

$$\mathrm{HBC}(i) = \sum_{s=1}^{s_{\max}} \frac{1}{s^2} \sum_{j \neq v_i \neq k}^{n} \frac{m_{jk}^s(v_i)}{m_{jk}^s}, \tag{22}$$

where m_{jk}^s denotes the number of shortest paths s between nodes v_j and v_k, and $m_{jk}^S(v_i)$ represents the number of these shortest paths that pass through v_i. This paper sets $s_{\max} = 3$.

5. Label propagation-based source identification algorithm (LPSI):[13] LPSI captures source centrality features in its methodology design. It proposes performing label propagation on the observed states of a network. By setting an initial state vector Y for iteration, the final convergence formula for label propagation is as follows:

$$Z^* = (1-\alpha)(1-\alpha S)^{-1} Y. \tag{23}$$

In the state vector Y, nodes in the infected or recovered state are set to 1, while all other nodes are set to -1. $S = D^{-1/2} A D^{-1/2}$ is the normalized weight matrix of the graph, and α is the score derived from label information from neighbors, with $\alpha = 0.5$ adopted in this paper. When a node's final label is greater than other nodes, it can be viewed as a source.

4.2. Datasets

Although the theoretical framework covers a higher-dimensional 3-simplex, the main experimental results focus on a 2-simplex for clarity. We ultimately validate through experiments that these findings generalize to a 3-simplex. Two synthetic network types were employed: random simplex complexes and scale-free simplex complexes.

The Erdős–Rényi simplicial complex (ERSC) model generates simplicial complexes with uniform degree distributions. The generation process is as follows: Generate edges between any two nodes with probability p_1. First, identify the existing 2-simplex in the generated random network. Then, generate a 2-simplex between any three nodes until the desired number of 2-simplex is reached.

The Barabási–Albert simplicial complex (BASC) model generates networks with heterogeneous degree distributions. The generation process is as follows: Generate a scale-free network using the Barabási–Albert model, where each newly added node preferentially selects m existing nodes to form edges. We first locate existing 2-simplex within the generated scale-free network, then generate 2-simplex between any three nodes. This continues until the desired number of 2-simplex is achieved. The generation of 3-simplex in ERSC and BASC follows the same procedure as 2-simplex, ensuring the closure structure of simplices throughout the process.

This study constructs real simplicial complexes using six real datasets. Among them, the InVS15, Thiers12, and Lyonschool simplicial complex networks are reconstructed from real-world face-to-face time-series datasets.[43] While Aves-weaver-social, Bio-diseasome, and Net-science are real simplicial complexes obtained by randomly selecting triangles from real networks[44] as 2-simplex. Table 1 presents the basic statistical characteristics of the real networks constructed using the aforementioned method. Here, N denotes the number of nodes in the network, $\langle k_1 \rangle$ represents the average degree of nodes in 1-simplex, $\langle k_1^2 \rangle$ represents the second-order average degree of nodes in 1-simplex, $k_2 = 3\,\mathrm{num}(\Delta)/N$ denotes the average degree of nodes in 2-simplex and $\beta_1^c = \frac{\langle k_1 \rangle}{\langle k_1^2 \rangle - \langle k_1 \rangle}$ is the threshold for the network propagation rate during an epidemic outbreak.

4.3. Experimental settings

To measure the performance of the algorithm, real source in rank is used to evaluate the performance of different algorithms in propagation source detection. This real source in rank is derived based on the calculated value of each algorithm. For example, the SCDMP algorithm sorts according to the energy values in Eq. (16) in ascending order, while the comparison algorithm sorts according to each centrality value in descending order. We randomly select 500 source nodes from V, simulate propagation for each, and average the detection results over all runs.

Table 2. Properties of simplicial complex networks.

Network	N	$\langle k_1 \rangle$	k_2	β_1^c
Aves-weaver-social	445	6	4.20	0.13
Bio-diseasome	516	4.60	3.16	0.13
Net-science	380	4.82	1.45	0.14
InVS15	211	7.52	2.19	0.11
Thiers12	156	4.56	1.21	0.19
Lyonschool	222	5.42	2.18	0.17
ERSC	500	3.92	0.98	0.24
BASC	500	3.97	1.08	0.16

4.4. Main results

First, we conduct experiments on a simplicial complex network with the highest dimension of 2, taking snapshot times $t = 4$ to 10. We observe the variation of the real source in the rank of each method across six real datasets and two synthetic datasets. As shown in Fig. 3, we find that our SCDMP algorithm consistently remains optimal in most cases, the LPSI algorithm performs sub-optimally, and the four centrality-based algorithms (JC, EIG, HBC, and HCC) perform the worst. At the snapshot time $t = 4$, the real source

rank of our algorithm is slightly higher than that of the comparison algorithms on seven out of the eight datasets, with the exception of Aves-weaver-social. This is because the snapshot time is too short, and the role of 2-simplex is relatively limited at this time, leading to poor performance. Moreover, at this time, all algorithms have similar source detection capabilities with little difference. Meanwhile, after $t = 4$, the real source ranking value of our algorithm is much lower than that of other algorithms, indicating that our algorithm is more effective. In addition, we observe that as the snapshot time increases and the propagation range expands, the source detection performance of each method deteriorates accordingly, which is consistent with real-world scenarios. When the infection range expands, it becomes more difficult for us to identify the source. Furthermore, among the six real datasets and two synthetic datasets, as the snapshot time becomes longer and detecting the real source becomes more difficult, the performance gap between our algorithm and other comparison algorithms becomes more obvious, indicating that our algorithm has higher stability and robustness.

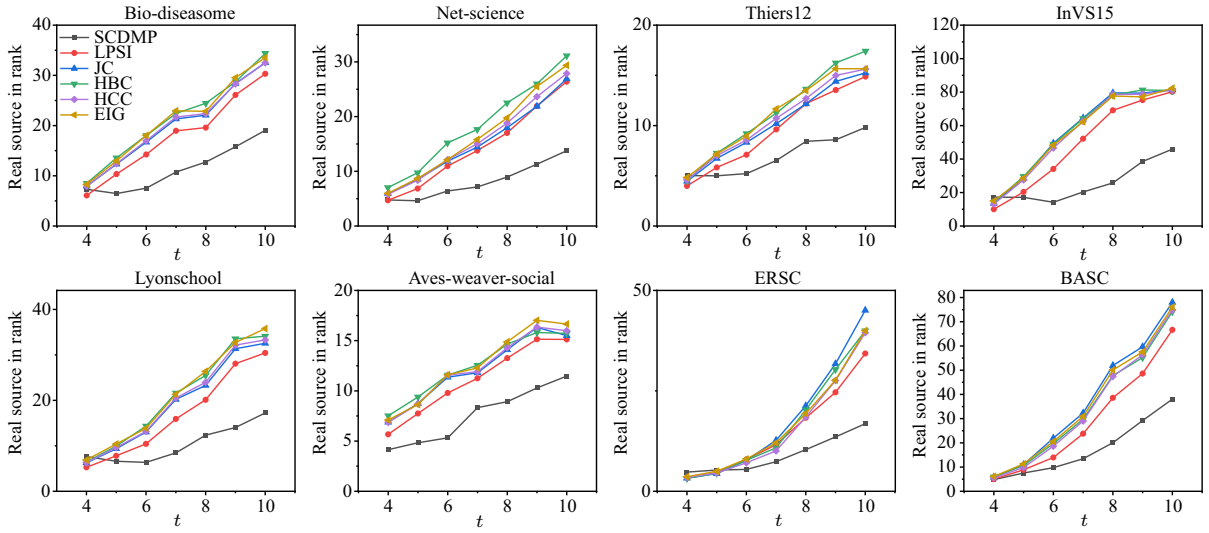


Fig. 3. Variation of the real source ranking values of different methods with snapshot time (2-simplex). $\beta_1 = 0.3$, $\beta_2 = 0.3$ and $\mu = 0.5$.

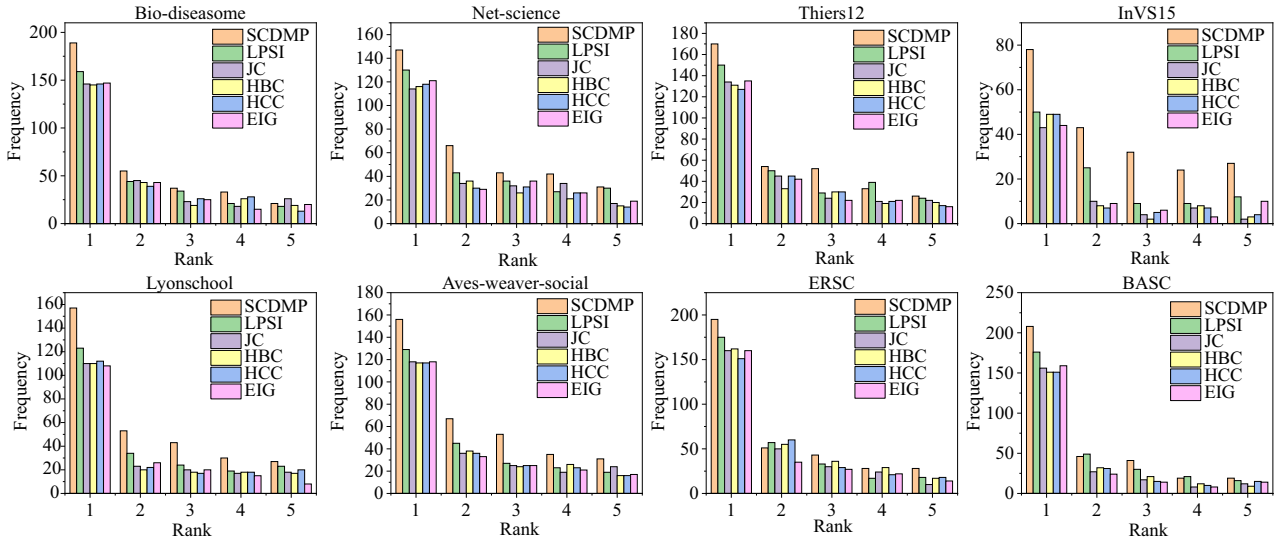


Fig. 4. Frequency distribution of the top 5 rankings of each method across eight datasets. $\beta_1 = 0.3$, $\beta_2 = 0.3$, and $\mu = 0.5$.

To demonstrate the accuracy of each method, we plot the frequency distribution of the top 5 rankings of each method under 500 simulations with parameters $t = 6$. As shown in Fig. 4, a ranking of 1 indicates that the predicted source node is the real node. Across the eight datasets, our algorithm outperforms all other comparison algorithms. Moreover, in most cases, when the real source ranks 2, 3, 4, or 5, our algorithm also outperforms the other algorithms, which is consistent with the experimental results of the average ranking of real source nodes. The LPSI algorithm performs sub-optimally, while the other four indicator-based algorithms perform similarly.

To further verify the performance of our algorithm under different propagation model parameters, we conduct a parameter sensitivity analysis on three propagation probability pa-

rameters (taking values from 0.1 to 0.8) across four different real datasets, with parameters $t = 6$. As shown in Fig. 5, under the fixed parameters $t = 6$, $\beta_2 = 0.3$, and $\mu = 0.5$, when taking different values of β_1, we find that as β_1 increases, the number of infected nodes increases, and the performance of all methods declines, which is consistent with real-world scenarios. In addition, we observe that as β_1 increases, the gap between the evaluation ranking of our algorithm and that of the comparison algorithms becomes larger, which corresponds to the conclusion in Fig. 3 and once again verifies the robustness of our algorithm. The second row of Fig. 5 presents the parameter variation experiment for the propagation probability of the 2-simplex. Due to the relatively small number of existing 2-simplex, the number of infected nodes does not increase significantly with the increase of β_2 as it does with β_1, so the change of comparison algorithms is not obvious. When $\beta_2 = 0.5$, the algorithm's performance declines due to its underlying assumption of strong 2-simplex effects. In practice, however, when 2-simplex are sparse, a higher β_2 does not substantially enhance their actual influence, leading the model to underperform compared to configurations with a lower β_2. Nevertheless, it still surpasses all baseline methods. The last row of Fig. 5 shows the parameter variation experiment for the recovery probability. From the experimental results, we can observe that as μ increases, the performance of all algorithms improves. This is because when the recovery probability becomes larger, the number of nodes in the infected state decreases, and thus the number of infected nodes also decreases, making it easier to identify the source node.

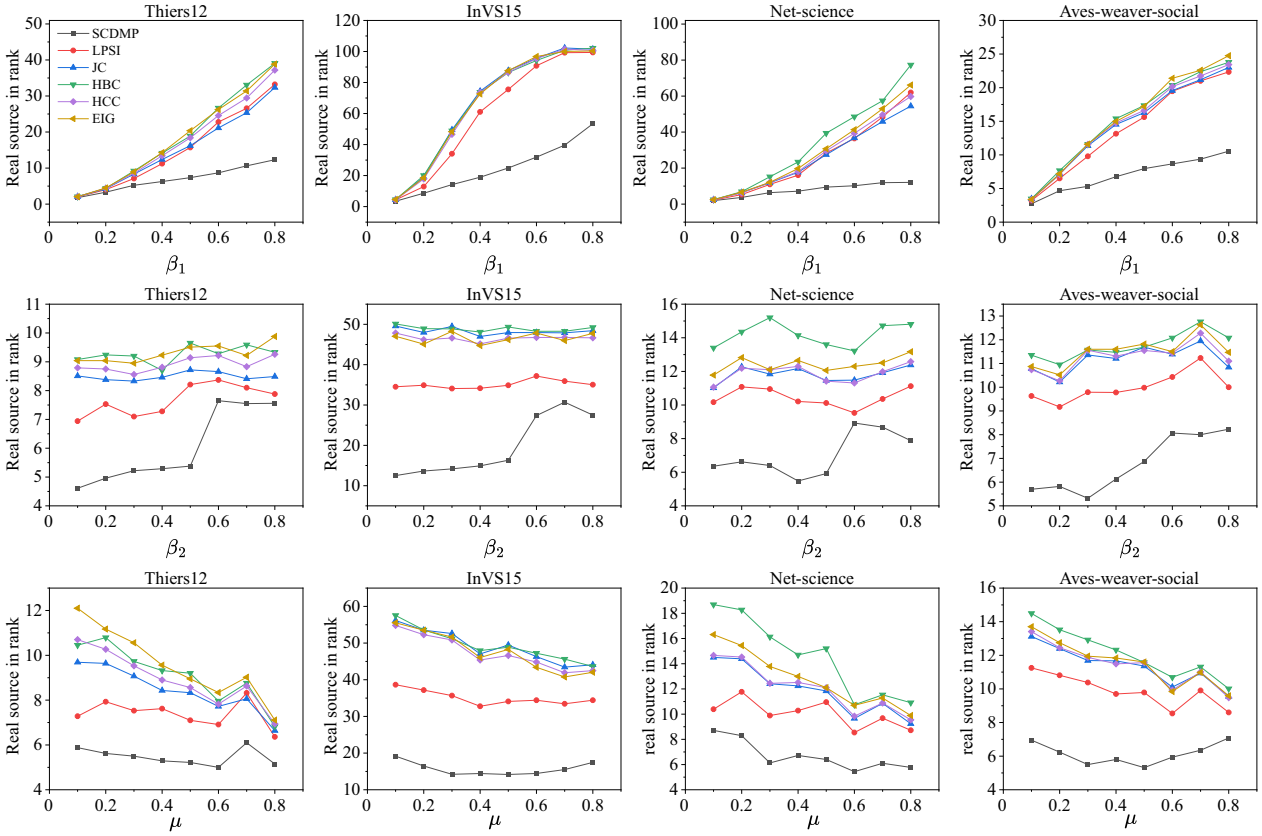


Fig. 5. Real source ranking values of various methods under different values of parameters β_1, β_2, and μ. Set $\beta_2 = 0.3$, $\mu = 0.5$ in the first line; set $\beta_1 = 0.3$, $\mu = 0.5$ in the second line; set $\beta_1 = 0.3$, $\beta_2 = 0.3$ in the third line.

To further validate the robustness of our algorithm, we conducted additional experiments under new parameter configurations. As shown in Fig. 6, the parameter settings are $(\beta_2, \mu) = (0.3, 0.8)$, $(\beta_1, \mu) = (0.3, 0.8)$, $(\beta_1, \beta_2) = (0.5, 0.3)$; and as shown in Fig. 7, the parameter settings are $(\beta_2, \mu) = (0.5, 0.5)$, $(\beta_1, \mu) = (0.5, 0.5)$, and $(\beta_1, \beta_2) = (0.3, 0.5)$. The results, presented in the newly added figures, exhibit similar trends to those previously observed, further confirming the consistency and stability of the proposed method.

In addition, since there are very few 3-simplex in real datasets, we generate two synthetic networks, ERSC3 and BASC3, with the highest simplex dimension of 3. As shown in Fig. 8, the accuracy of real source ranking for different methods is presented across varying network sizes (columns, from small to large) on ERSC (first row) and BASC (second row) synthetic networks. Experimental results show that as the network size increases, the trends of real source ranking over snapshot time exhibit varying degrees of fluctuation across different methods. However, under all size settings, our proposed method consistently outperforms the compared methods in terms of ranking accuracy and stability. Especially in

large-scale networks, where the ranking performance of other methods significantly degrades, our method maintains a high ranking accuracy, demonstrating its excellent scalability and robustness.

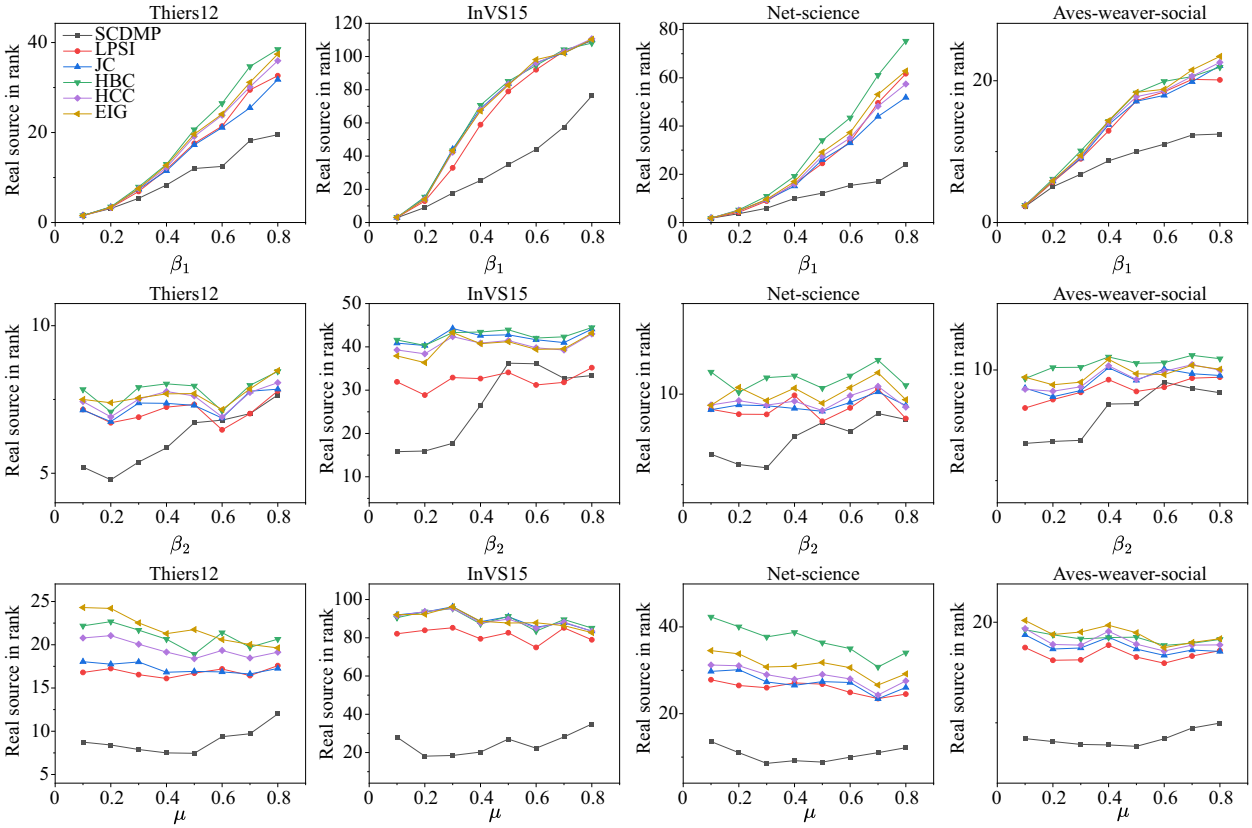


Fig. 6. Real source ranking values of various methods under different values of parameters β_1, β_2, and μ. Set $\beta_2 = 0.3$, $\mu = 0.8$ in the first line; set $\beta_1 = 0.3$, $\mu = 0.8$ in the second line; set $\beta_1 = 0.5$, $\beta_2 = 0.3$ in the third line.

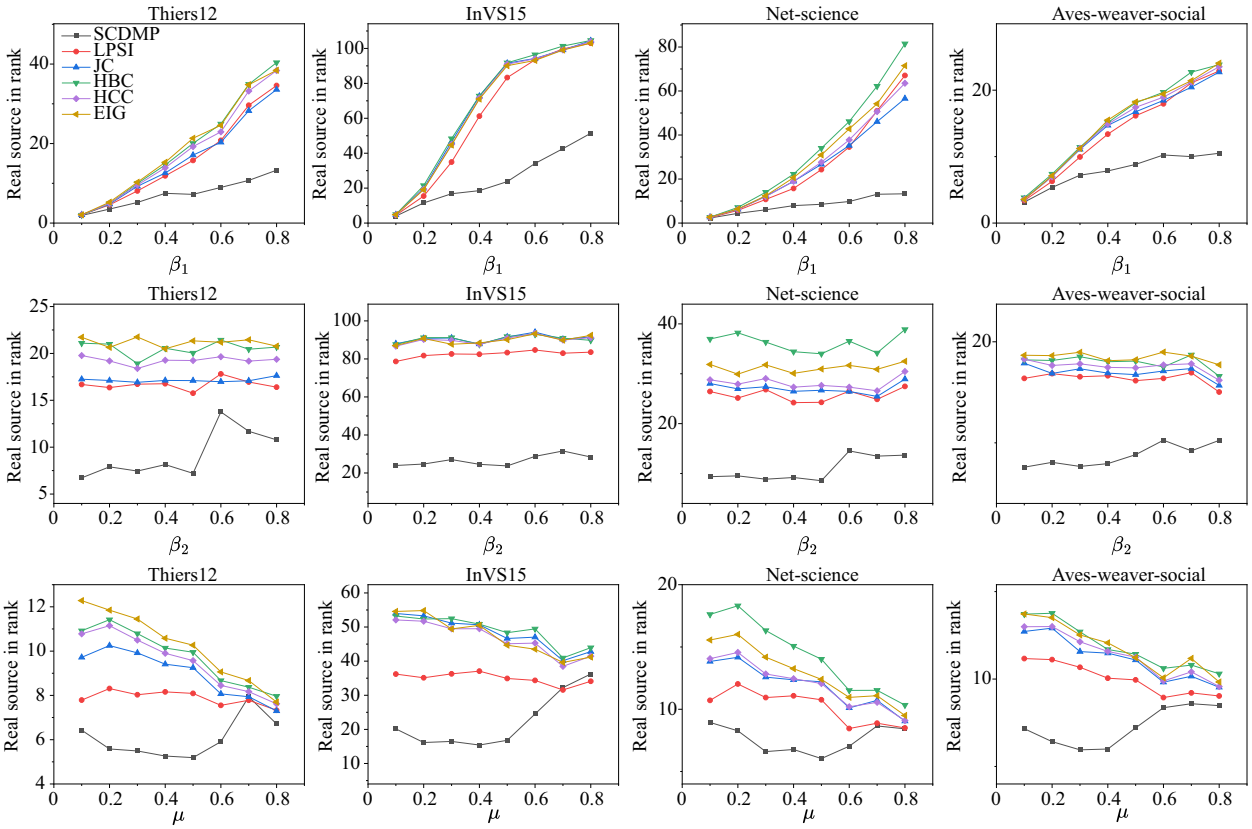


Fig. 7. Real source ranking values of various methods under different values of parameters β_1, β_2, and μ. Set $\beta_2 = 0.5$, $\mu = 0.5$ in the first line; set $\beta_1 = 0.5$, $\mu = 0.5$ in the second line; set $\beta_1 = 0.3$, $\beta_2 = 0.5$ in the third line.

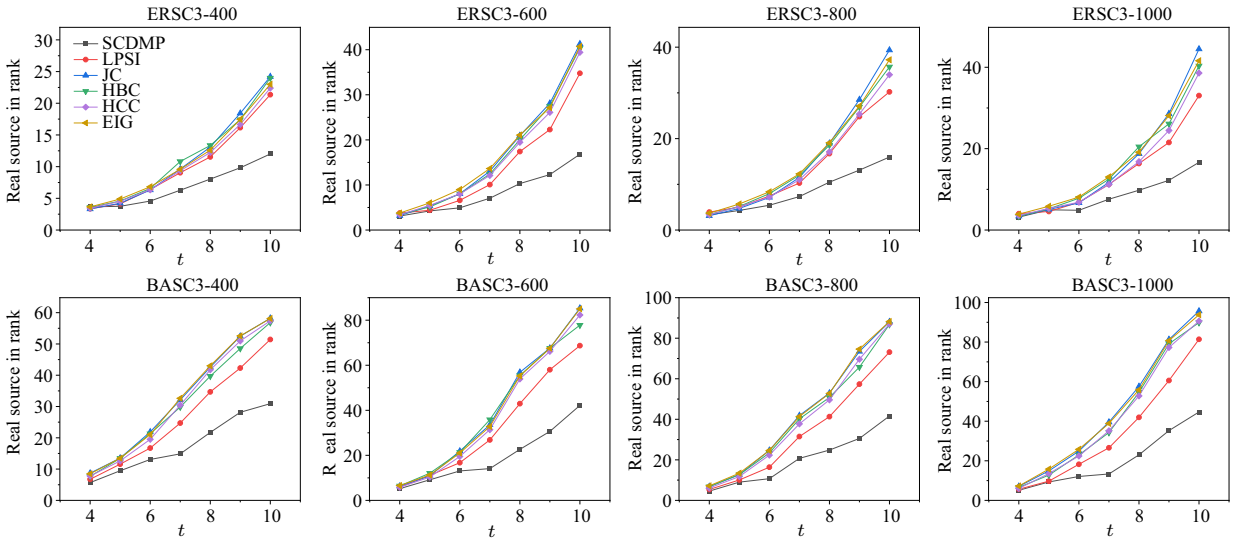


Fig. 8. Real source ranking of different algorithms on ERSC and BASC synthetic networks of varying sizes (400, 600, 800, 1000). $\beta_1 = 0.3$, $\beta_2 = 0.3$, and $\mu = 0.5$.

5. Conclusion

The accurate identification of information sources in complex networks is pivotal for tasks such as public opinion monitoring and epidemic control. This paper studies source detection problems using a dynamic message passing algorithm defined on simplicial complexes. Specifically, we first introduce an SIR spreading model on simplicial complexes, then develop a corresponding message passing framework. Building upon this theoretical foundation, we subsequently develop an energy-based metric to infer the propagation source within simplicial complex networks. Experiments on two synthetic datasets and six realistic datasets, in comparison with five baseline methods, are conducted under varying simulation conditions (including different snapshot times, simplex propagation probabilities, and recovery probabilities) to comprehensively evaluate performance. The results show that the proposed SCDMP algorithm consistently achieves the best performance in most scenarios. Whether on simplicial complexes of maximum dimension 2 or 3, the performance advantage of SCDMP becomes increasingly pronounced as snapshot times grow and the source localization problem becomes more challenging, demonstrating the algorithm's stability and robustness. Furthermore, parameter sensitivity analyses confirm that SCDMP maintains strong performance across diverse model settings.

Although the SCDMP algorithm demonstrates promising performance in source detection tasks, several aspects warrant further investigation. Future work could extend the current framework to multi-source scenarios, which are more common in real-world propagation such as multi-wave epidemics or rumors originating from multiple sources. In addition, it would also be valuable to explore the integration and comparison of the proposed method with other higher-order network representations. Such efforts would advance the theoretical development of source detection in complex networks and support its broader application in real-world propagation settings.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62473001 and 62503006), the Natural Science Foundation of Anhui Province (Grant No. 2408085QF210), and the High-level Talent Introduction Project of Anhui University of Chinese Medicine (Grant No. DT2400000526).

References

[1] Zhang C, Guo Q, Fu L, Gan X and Wang X 2021 *IEEE Infocom* **2021** 1
[2] Paluch R, Gajewski Ł Suchecki K, Szymański B and Hołyst J A 2020 *Springer Proc. Complex.* pp. 21–41
[3] Yang F, Yang S, Peng Y, Yao Y, Wang Z, Li H, Liu J, Zhang R and Li C 2020 *Knowl. Based Syst.* **195** 105674
[4] Wang Z, Hou D, Gao C, Huang J and Xuan Q 2022 *Proc. ACM Web Conf.* **2022** 1372
[5] Jalili M and Perc M 2017 *J. Complex Netw.* **5** 665
[6] Comin C H and da Fontoura Costa L 2011 *Phys. Rev. E* **84** 056105
[7] Jiang J, Wen S, Yu S, Xiang Y and Zhou W 2016 *IEEE Commun. Surv. Tutor.* **19(1)** 465
[8] Ali S S, Anwar T and Rizvi S A M 2020 *Online Soc. Netw. Media* **17** 100061
[9] Shah D and Zaman T 2010 *Proc. ACM Sigmetrics* **2010** 203
[10] Shah D and Zaman T 2011 *IEEE Trans. Inf. Theory* **57** 5163
[11] Luo W, Tay W P and Leng M 2014 *IEEE J. Sel. Top. Signal Process.* **8** 586
[12] Zhu K and Ying L 2014 *IEEE/ACM Trans. Netw.* **24** 408
[13] Wang Z, Wang C, Pei J and Ye X 2017 *Proc. AAAI Conf. Artif. Intell.* **31** 217
[14] Deng Y Z, Hu Z L, Lin F, Tang B, Wang H and Huang Y 2024 *Chin. Phys. B* **33** 118901
[15] Altarelli F, Braunstein A, Dall'Asta L, Lage-Castellanos A and Zecchina R 2014 *Phys. Rev. Lett.* **112** 118701
[16] Lokhov A Y, Mézard M, Ohta H and Zdeborova L 2014 *Phys. Rev. E* **90** 012801
[17] Bagaev D and de Vries B 2023 *Sci. Program.* **2023** 6601690
[18] Yu X, Nie Y, Li W, Lin T, Chen Y, Gao F and Wang W 2024 *Int. J. Mod. Phys. C* **35** 2450146
[19] Jiang Z X, Hu Z L and Huang F 2024 *Chaos Solitons Fractals* **188** 115532

[20] Shi C, Zhang Q and Chu T 2022 *Chin. Phys. B* **31** 070203
[21] Ma Z W, Sun L, Ding Z G, Huang Y and Hu L 2024 *Chin. Phys. B* **33** 028902
[22] Zhao J and Cheong K H 2023 *Inf. Sci.* **642** 119061
[23] Chang B, Chen E, Zhu F, Liu Q, Xu T and Wang Z 2020 *IEEE Trans. Syst. Man Cybern. Syst.* **50** 2242
[24] Hu Z L, Wang H J, Sun L, Tang B and Li M 2024 *Expert Syst. Appl.* **256** 124946
[25] Cheng L, Zhu P, Tang K, Gao C and Wang Z 2025 *IEEE Trans. Inf. Forensics Security* **20** 1
[26] Shu X, Yu B, Ruan Z, Zhang Q and Xuan Q 2021 *Graph Data Min.* **1** 1
[27] Ran H, Jia C, Zhang P and Li X 2022 *Inf. Sci.* **592** 402
[28] Čutura G, Li B, Swami A and Segarra S 2021 *Proc. Eur. Signal Process. Conf.* **2021** 2204
[29] He Q, Yan X, Fang H, Chen L and Zhang J 2025 *IEEE/ACM Trans. Netw.* **33** 1
[30] Chen J H, Zou K, Zhang L M and Huang C W 2025 *Europhys. Lett.* **151** 41001
[31] Yu X, Nie Y, Li W, Luo G, Lin T and Wang W 2024 *Chaos Solitons Fractals* **187** 115457
[32] He Q, Bao Y, Fang H, Lin Y and Sun H 2025 *Proc. AAAI Conf. Artif. Intell.* **39** 291
[33] Liang L, Qi L Y and Gong S C 2025 *Chin. Phys. B* **34** 108904
[34] Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, Young J G and Petri G 2020 *Phys. Rep.* **874** 1
[35] Iacopini I, Petri G, Barrat A and Latora V 2019 *Nat. Commun.* **10** 2485
[36] Fan X N and You X M 2025 *Chin. Phys. B* **34** 038703
[37] Centola D and Macy M 2007 *Am. J. Sociol.* **113** 702
[38] Su Z, Chen L, Ai J, Zheng Y Y and Bie N 2024 *Chin. Phys. B* **33** 058901
[39] Qu H, Song Y R, Li R, Li M, and Jiang G P 2025 *Chin. Phys. B* **34** 028901
[40] Xu Q, Sun L and Bu C 2023 *Chaos Solitons Fractals* **173** 113753
[41] Gao Z, Shi Y and Chen S 2015 *Int. J. Mod. Phys. C* **26** 1550107
[42] Xie X, Zhan X, Zhang Z and Liu C 2023 *Chaos* **33** 013101
[43] Wang H, Ma C, Chen H S, Lai Y and Zhang H 2022 *Nat. Commun.* **13** 3043
[44] Rossi R and Ahmed N 2015 *Proc. AAAI Conf. Artif. Intell.* **29(1)** 1