

Identification of vital nodes based on global and local features in hypergraphs

Li Liang(梁丽), Li-Yao Qi(齐丽瑶), and Shi-Cai Gong(龚世才)

Citation: Chin. Phys. B, 2025, 34 (**10**): 108904. DOI: 10.1088/1674-1056/adfefc
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

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

Identifying important nodes of hypergraph: An improved PageRank algorithm

Yu-Hao Piao(朴宇豪), Jun-Yi Wang(王俊义), and Ke-Zan Li(李科赞)
Chin. Phys. B, 2025, 34 (**4**): 048902. DOI: 10.1088/1674-1056/adb269

Vital nodes identification method integrating degree centrality and cycle ratio

Yu Zhao(赵玉) and Bo Yang(杨波)
Chin. Phys. B, 2025, 34 (**3**): 038901. DOI: 10.1088/1674-1056/ada42d

Critical station identification of metro networks based on the integrated topological-functional algorithm: A case study of Chengdu

Zi-Qiang Zeng(曾自强), Sheng-Jie He(何圣洁), and Wang Tian(田旺)
Chin. Phys. B, 2025, 34 (**2**): 028902. DOI: 10.1088/1674-1056/ad9734

Influence fast or later: Two types of influencers in social networks

Fang Zhou(周方), Chang Su(苏畅), Shuqi Xu(徐舒琪), and Linyuan Lü(吕琳媛)
Chin. Phys. B, 2022, 31 (**6**): 068901. DOI: 10.1088/1674-1056/ac4484

Detection of influential nodes with multi-scale information

Jing-En Wang(王静恩), San-Yang Liu(刘三阳), Ahmed Aljmiai, and Yi-Guang Bai(白艺光)
Chin. Phys. B, 2021, 30 (**8**): 088902. DOI: 10.1088/1674-1056/abff2d

---

SPECIAL TOPIC — Computational programs in complex systems

Identification of vital nodes based on global and local features in hypergraphs

Li Liang(梁丽), Li-Yao Qi(齐丽瑶), and Shi-Cai Gong(龚世才)[†]

School of Science, Zhejiang University of Science and Technology, Hangzhou 310023, *China*

(Received 24 June 2025; revised manuscript received 4 August 2025; accepted manuscript online 26 August 2025)

Hypergraphs, which encapsulate interactions of higher-order beyond mere pairwise connections, are essential for representing polyadic relationships within complex systems. Consequently, an increasing number of researchers are focusing on the centrality problem in hypergraphs. Specifically, researchers are tackling the challenge of utilizing higher-order structures to effectively define centrality metrics. This paper presents a novel approach, LGK, derived from the K-shell decomposition method, which incorporates both global and local perspectives. Empirical evaluations indicate that the LGK method provides several advantages, including reduced time complexity and improved accuracy in identifying critical nodes in hypergraphs.

Keywords: hypergraph, vital nodes, K-shell decomposition, susceptible–infected–recovered (SIR) model

PACS: 89.75.–k, 89.75.Fb, 05.45.–a

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

1. Introduction

With the rapid development of information technology and improved data acquisition capabilities, complex networks have become vital analytical tools.[1–5] In such networks, the loss or activation of critical nodes can significantly impact system stability and operational efficiency. Identifying these nodes is crucial for effective resource allocation, disease transmission mitigation, and rumor propagation management.[6–12] Consequently, the centrality problem underpins node importance identification and has become a central focus in complex network research. Researchers have proposed various centrality measures[13–17] from both structural and dynamic perspectives to identify vital nodes.

As network complexity and scale expand, the limitations of traditional networks grow increasingly evident. Hypergraphs, capable of representing higher-order interactions within complex systems, offer significant potential for applications such as paper collaboration networks,[18] protein interaction networks,[19] and face-to-face interaction networks.[20] However, existing methods for identifying key nodes in traditional networks often struggle to effectively capture the true influence of nodes in hypergraph environments. This difficulty arises from the unique high-order structure of hypergraphs, which not only allows for pairwise interactions between nodes but also facilitates high-order interactions involving multiple nodes. To overcome these challenges, researchers have proposed a series of key node identification methods that take high-order structures into account.

In 2013, Kapoor[21] introduced the concept of hyperdegree, which refers to the number of hyperedges linked to a given node. In 2015, Vogiatzis *et al.*[22] extended the traditional notion of betweenness centrality to hypergraphs to identify influential groups of a fixed size. Subsequently, in 2019, Benson[23] introduced the eigenvector centrality as a method to assess the relative importance of nodes in uniform hypergraphs, highlighting the critical role played by a node's neighbors. In 2022, Kovalenko *et al.*[24] developed the notion of vector centrality by amalgamating eigenvector values of nodes calculated across various hyperedge cardinalities, thereby providing a framework for assessing node importance across differing hyperedge sizes. In 2023, Zeng *et al.*[25] proposed a higher-order enhanced random walk method by designing a random walk process on a constructed multi-order network model, effectively mitigating the localization issue inherent in traditional random walk approaches. In 2025, Piao *et al.*[26] introduced an improved PangRank algorithm for identifying influential nodes in directed hypergraphs, leveraging the Jaccard similarity coefficient to evaluate the proportion of shared neighbors between nodes.

However, the aforementioned centrality metrics primarily assess node importance either from a local, a semi-global or a global perspective, resulting in a lack of comprehensiveness. In 2019, Hu *et al.*[27] identified critical proteins within the protein complex hypernetwork by utilizing complex parameters, including hyperdegree distribution, degree distribution, and sub-hypergraph centrality.[28] Although this approach mitigates the limitations of single-parameter assessments by integrating multiple node characteristics, it fails to account for the

[†]Corresponding author. E-mail: gongsc@zust.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

weighting of these parameters, which may lead to potentially inaccurate evaluations. In 2021, Zhou *et al.*[29] adapted the K-shell decomposition method from traditional networks for application in hypergraphs, proposing an index for identifying critical nodes in hypergraphs. This index integrates hyperdegree with the K-shell decomposition value through the Euclidean distance formula to improve the drawback of insufficient discrimination when using only K-shell decomposition. Hu *et al.*[30] introduced an improved method of identifying vital nodes in 2023, which is based on the entropy method and extends the K-shell decomposition for hypergraphs. This method evaluates node influence from both local and global perspectives, considers the effects of neighbors, and adaptively assigns weights to each index through the application of entropy theory. In the same year, Xie *et al.*[31] introduced a hypergraph centrality measure based on the gravity model, termed gravity-based centrality in hypergraphs (HGC). This approach evaluates node quality by considering their degrees and employs higher-order distance to represent the distance between nodes, based on the principle that objects with greater mass and closer proximity exert a stronger gravitational influence. However, while this method enhances recognition accuracy, it also significantly increases computational complexity. In 2024, Wang *et al.*[32] introduced a method to identify vital nodes in multilayer hypergraphs, incorporating local, semi-local, and global perspectives. In 2025, Guo *et al.*[33] applied the h-index to the weighted aggregated hypernetwork, integrating it with the K-shell algorithm and hyperdegree metric to identify vital nodes.

Although existing methods have to some extent alleviated the limitations caused by relying on a single metric or perspective, two significant challenges remain. First, in K-shell-based centrality measures, the differentiated contributions of neighbors located at different K-shell levels to a node's influence are often neglected. Second, it remains difficult to simultaneously achieve a favorable trade-off between computational complexity and identification accuracy. Therefore, a critical open problem in hypergraph-based node importance identification is how to effectively integrate nodes and the multi-level structural features of their neighbors to design a method that ensures high accuracy while maintaining computational efficiency.

Based on the above research motivations and practical needs, this paper proposes a novel method for identifying influential nodes in hypergraph, termed LGK (local–global K-shell) method, which integrates both global and local structural features. Specifically, the core innovation of LGK lies in the introduction of the combination hyperdegree (denoted as DK), which fuses the node's own K-shell information with a weighted contribution of its neighbors across different shell levels based on their respective participation in hyperedges. Furthermore, the DK of global perspective is combined with the hyperdegree of local perspective through Min–Max normalization, ensuring a more balanced contribution to the final importance score. This results in the LGK index, which more accurately captures a node's actual influence in spreading processes. Experimental results demonstrate that LGK not only achieves higher accuracy in identifying vital nodes, but also exhibits superior computational efficiency.

Notably, transmission dynamics provides a robust theoretical framework for the process evaluating the effectiveness of vital node identification methods. By simulating the spreading process of diseases or information in networks, it assesses the efficacy of these methods in identifying vital nodes for controlling or predicting the propagation process. In recent years, epidemic spreading dynamics[34] on high-order networks have attracted considerable attention, with researchers developing various models to simulate both single and interacting epidemic spreading dynamics. In 2018, Suo *et al.*[35] constructed two SIS-based spreading models that incorporate distinct transmission strategies: the reactive process and the contact process. These models integrate both the spreading strategy and network structure to reflect the diversity of real-world diffusion scenarios. In 2020, Landry *et al.*[36] investigated epidemic spreading on hypergraphs using the SIS model framework, emphasizing the impact of network heterogeneity on contagion dynamics. In the same year, Ran *et al.*[37] proposed a generalized linear threshold (GLT) model to address the shortcomings of the traditional linear threshold (LT) model in system evolution studies. In 2022, Nie *et al.*[38] proposed a competing behavior spreading model grounded in the s-SIS mechanism, which considers higher-order interactions, population heterogeneity, and homophily effects. In the same year, St-Onge *et al.*[39] presented a nonlinear hypergraph contagion framework, which better adapted to the complexity of real-world scenarios and yielded more accurate predictions regarding propagation dynamics. In 2023, Zhang *et al.*[40] proposed a hyper-SIR rumor spreading model based on hypergraph theory, which captures the influence of higher-order interactions on rumor dynamics and provides a more precise representation of the rumor diffusion process. In 2025, Cui *et al.*[41] proposed a single-virus SIWS model based on directed hypergraph by combining direct pathway and indirect pathway. In the same year, Zhou *et al.*[42] developed a susceptible–infected–recovered (SIR) information-spreading model that incorporates social reinforcement, using both real and synthetic higher-order networks.

Most of the existing models are extensions based on the classical SIS or SIR frameworks. Among them, the SIR model explicitly defines three epidemiological states — susceptible, infected, and recovered — making it particularly suitable for simulating realistic contagion processes such as disease or in-

formation diffusion. Furthermore, nonlinear variant of the SIR model possesses theoretical and practical advantages over the linear counterpart. Specifically, nonlinear contagion models account for complex propagation characteristics induced by multi-body interactions, such as superspreading events or social reinforcement, thereby aligning more closely with real-world spreading scenarios. Moreover, such models are capable of producing richer dynamical behaviors, including discontinuous phase transitions and bistability, that better capture the diversity of outcomes observed in real contagion processes. Consequently, in this study, we utilize the SIR nonlinear propagation model as a framework for evaluating the influence of nodes in hypergraphs.

The subsequent sections of this paper are structured as follows. Section 2 provides an overview of key concepts and baseline methods. Section 3 describes the LGK method in detail. Section 4 presents the experimental findings and their subsequent analysis. Finally, Section 5 offers a concise summary of the conclusions of the study. Table 1 lists some key symbols and definitions used in this paper.

Table 1. Notations and definitions.

Notation	Definition
N	The number of nodes
M	The number of hyperedges
k_i	The degree of v_i
s_{m}	The maximum cardinality
HDC	Hyperdegree centrality
CC	Closeness centrality
BC	Betweenness centrality
k_s^{d}	The metric in Ref. [29]
GI	The metric in Ref. [29]
HGC	The method in Ref. [31]
LGK	The method introduced in this paper
$N_{\mathrm{h}}^{j}(v_i)$	The combination hyperdegree introduced in this paper
$k_s(v_i)$	The k-shell decomposition value of v_i
$d^{\mathrm{H}}(i,j)$	The higher-order distance introduced in Ref. [31]
$\beta(h,i)$	The probability of a node being infected in the SIR model
γ	The probability of node recovery in the SIR model
ϵ_s	The network efficiency of hypergraph H

2. Preliminaries

2.1. Hypergraph

Hypergraphs, as a generalization of traditional graphs, allow edges to connect multiple nodes simultaneously. This characteristic significantly enhances their capacity to represent complex scenarios more effectively. The notion of hypergraphs was initially introduced by Berge in the 1960s.[43,44] A hypergraph, denoted as H, consists of two components: the node set $V = \{v_1, v_2, \ldots, v_N\}$ and the hyperedge set $E = \{e_1, e_2, \ldots, e_M\}$. Each element e_j (where $1 \le j \le M$) in E is characterized as a nonempty subset of V, referred to as a hyperedge. When each nonempty subset in E contains exactly two nodes, H simplifies to a traditional graph, thereby illustrating that traditional graphs represent a specific instance of hypergraphs.

Figure 1 illustrates a fundamental example of a hypergraph. The node set of the hypergraph is $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$, and the hyperedge set is $E = \{e_1, e_2, e_3, e_4\}$. Specifically, $e_1 = \{v_1, v_2\}$, $e_2 = \{v_2, v_3, v_4\}$, $e_3 = \{v_4, v_5, v_6\}$, and $e_4 = \{v_1, v_6, v_7\}$.

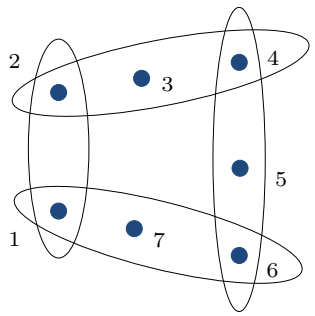


Fig. 1. A simple hypergraph with 7 nodes and 4 hyperedges.

2.2. Baseline methods

Taking into account both complexity and accuracy comprehensively, we have selected five baseline methods, including the hyperdegree centrality, the closeness centrality, the betweenness centrality, the gravity-based centrality in hypergraphs, the k_s^{d} metric, and the *GI* metric, which will be introduced in this section.

2.2.1. Hyperdegree centrality (HDC)

Hyperdegree centrality[21] is a direct measure of the number of hyperedges involved by nodes in a hypergraph. The HDC score of v_i can be obtained by

$$\mathrm{HDC}(v_i) = \sum_{j=1}^{M} b_{ji}, \tag{1}$$

where b_{ji} denotes an element in the incidence matrix of the hypergraph, and it illustrates the connection situation between the node v_i and the hyperedge e_j.

2.2.2. Closeness centrality (CC)

The closeness centrality[45] highlights a node's accessibility to other nodes in a hypergraph. A higher closeness centrality indicates a greater efficiency in information dissemination. The CC value of v_i can be obtained by

$$\mathrm{CC}(v_i) = \frac{n-1}{\sum_{i \neq j} d_{ij}}, \tag{2}$$

where d_{ij} denotes the shortest distance between v_i and v_j.

2.2.3. Betweenness centrality (BC)

The betweenness centrality[22] is a global metric indicating the significance of a node as a "bridge". The BC value of

v_i can be obtained by

$$\mathrm{BC}(v_i) = \frac{\sum\limits_{\substack{1\le j<k\le N \\ j\neq k\neq i}} \frac{g_{kj}(v_i)}{g_{kj}}}{\frac{N(N-1)}{2}}, \tag{3}$$

where g_{kj} denotes the number of shortest paths between v_j and v_k, and $g_{kj}(v_i)$ represents the number of these paths that incorporate v_i.

2.2.4. Gravity-based centrality in hypergraphs (HGC)

The gravity-based centrality in hypergraphs[31] is derived from the gravity model, which posits that nodes with higher degrees and shorter distances exhibit stronger mutual attraction. The HGC score of v_i can be obtained by

$$\mathrm{HGC}(v_i) = \sum_{\substack{i,j \\ i\neq j}} \frac{k_i \cdot k_j}{(d^{\mathrm{H}}(i,j))^2}, \tag{4}$$

where k_i and k_j denote the degrees of v_i and v_j, respectively, which can be obtained based on the adjacency matrix, and $d^{\mathrm{H}}(i,j)$ represents the higher-order distance between v_i and v_j, and its specific formulation is as follows:

$$d^{\mathrm{H}}(i,j) = \sum_{s=1}^{s_{\mathrm{m}}} \frac{d_s^v(i,j)}{s^2}, \tag{5}$$

where $d_s^v(i,j)$ denotes the s-distance between v_i and v_j. When v_i and v_j are adjacent, $d_s^v(i,j)$ equals 1; otherwise, it equals $d_s^e(g,q)+1$. The $d_s^e(g,q)$[45] denotes the s-distance between e_g and e_q, where v_i and v_j belong to e_g and e_q, respectively.

2.2.5. Based on K-shell decomposition – k_s^{d} and GI

The k_s^{d}[29] integrates the k_s with the hyperdegree. The k_s^{d} value of v_i can be obtained by

$$k_s^{\mathrm{d}}(v_i) = \sqrt{\mathrm{HDC}^2(v_i) + k_s^2(v_i)}. \tag{6}$$

The GI[30] integrates the k_s values of both individual nodes and their neighbors. The GI score of v_i can be obtained by

$$\mathrm{GI}(v_i) = k_s(v_i) + \sum_{v_j\in\Gamma(v_i)} k_s(v_j), \tag{7}$$

where $\Gamma(v_i)$ represents the set of neighbors of node v_i.

2.3. Hypergraph contagion model and network efficiency

2.3.1. SIR nonlinear propagation model

In the SIR nonlinear propagation model, a node can exist in one of three distinct states: susceptible (S), infected (I), or recovered (R). The infection process is detailed below.

1. Initialization. Select a subset of nodes as the initial I-state nodes.

2. Contagion. At each time step, the S-state node linked solely to one hyperedge becomes infected with a probability $\beta(h,i)$, defined as follows:

$$\beta(h,i) = \alpha \cdot i^{\kappa}, \tag{8}$$

where κ is an adjustable parameter, while h represents the cardinality of the hyperedge, and i represents the number of I-state nodes in the hyperedge. Note that $\alpha = \Lambda(h)$. In this paper, we deliberately selected the infection rate function independent of h in order to emphasize the impact of the nonlinearity.

3. Recovery. At each time step, the I-state nodes recover to the R-state with a probability γ independently.

4. Termination. Repeat steps 2 and 3 until there are no remaining I-state nodes in the hypergraph.

Figure 2 illustrates the SIR spreading process. In a hyperedge with cardinality h, if i nodes (where $i \le h$) are in the I-state, then each of the remaining $(h-i)$ S-state nodes in the hyperedge is infected with a probability $\beta(h,i)$. For S-state nodes linked to multiple hyperedges, their total probability of becoming infected equals the cumulative sum of the infection probabilities from each hyperedge they are linked to. Furthermore, I-state nodes transition back to the R-state with a probability γ.

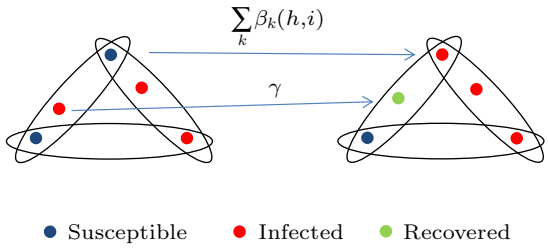


Fig. 2. SIR nonlinear propagation model in hypergraph. The S-state node (blue circles) becomes infected with a probability $\sum_k \beta_k(h,i)$, and the I-state node (red circles) recovers with a probability γ. Take the topmost node, which becomes infected with a probability $\beta_1(3,1)+\beta_2(3,2)$.

2.3.2. Network efficiency

The spreading model elucidates the significance of a node based on its dynamic characteristics, while network efficiency functions as an indicator of a node's functional significance. Network efficiency is a quantitative measure that assesses the efficacy of transmission within a network, positing that transmission becomes less efficient as the distance between nodes increases. In hypergraphs, the transmission efficiency between hyperedges is converted into the transmission efficiency among nodes in ordinary graphs by constructing a line graph, thereby enabling the computation of transmission efficiency.[31] Considering that transmission efficiency between hyperedges increases with the number of shared nodes, the hypergraph network efficiency is defined as the cumulative sum ϵ_s to capture the aggregate impact of overlaps of size s on overall network resilience. It is calculated as

follows:

$$\epsilon_s = \sum_{s=1}^{s'} \sum_{\substack{e_g, e_q \in E_s \\ g \neq q}} \frac{2}{|E_s|(|E_s|-1)\cdot d_s^e(g,q)}, \tag{9}$$

where $E_s = \{e_m : |e_m| \geq s\}$ represents a set of hyperedges that include a minimum of s nodes, and $d_s^e(g,q)$ is the s-distance[45] between e_g and e_q.

3. Proposed method

In this section, we propose an innovative approach for identifying critical nodes in hypergraphs, termed the LGK method. This method is based on the K-shell decomposition and integrates both global and local perspectives.

3.1. K-shell decomposition in hypergraphs

The K-shell decomposition[46] is a widely utilized technique in network analysis for assessing the structural significance of nodes. It decomposes the network into hierarchical layers, or shells, grouping nodes with similar connectivity patterns. The concept of k_s for hypergraphs extends the K-shell decomposition approach applied to traditional networks, thereby enabling the evaluation of the positional importance of nodes in a hypergraph.

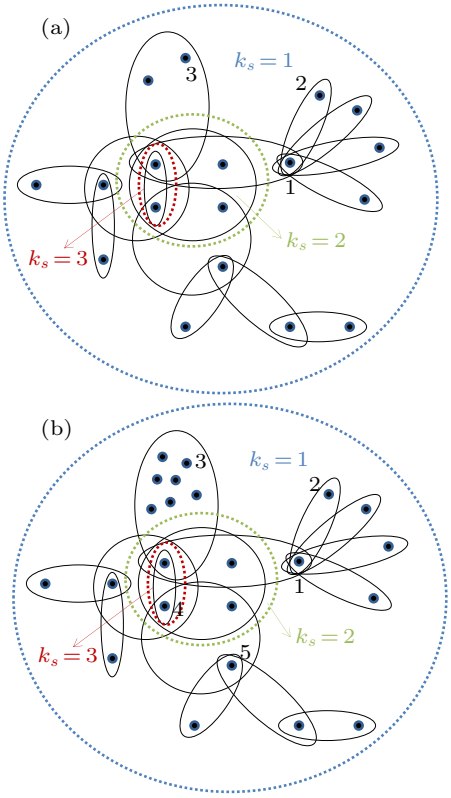


Fig. 3. K-shell decomposition diagram of hypergraph.

The computation of the k_s value involves a series of systematic steps. Initially, nodes with a hyperdegree of 1 and hyperedges with a cardinality of 1 are removed. This procedure is repeated until no such nodes remain. Each eliminated node is subsequently assigned a k_s value of 1. The process is then repeated to exclude all nodes with a hyperdegree of 2, along with hyperedges of cardinality 1, resulting in the assignment of a k_s value of 2 to all nodes removed during this phase. This iterative process continues until every node in the hypergraph has been assigned a k_s value. A higher k_s value of a node indicates a more central position in the hypergraph. It is noteworthy that a node located at the center of a hypergraph may also possess a relatively low hyperdegree. The K-shell decomposition diagram of the hypergraph is presented in Fig. 3.

3.2. LGK method

Essentially, a node's significance is not only related to the value of k_s, but also to the state of its neighboring nodes. Thus, we define a combination hyperdegree DK of node v_i as

$$\mathrm{DK}(v_i) = k_s(v_i) \sum_{j=1}^{K} \left(j \times N_{\mathrm{h}}^{j}(v_i) \right), \tag{10}$$

where K represents maximum value of k_s, $N_{\mathrm{h}}^{j}(v_i)$ represents the number of hyperedges participated by v_i's neighbors whose k_s value is j.

The combination hyperdegree based on K-shell decomposition serves as a global metric. To assess node importance comprehensively, we further integrate this global metric with the most fundamental local metric hyperdegree. Consequently, the definition of the vital node centrality metric LGK is

$$\mathrm{LGK}(v_i) = \mathrm{HDC}^{\mathrm{n}}(v_i) + \mathrm{DK}^{\mathrm{n}}(v_i). \tag{11}$$

Significantly, $\mathrm{HDC}^{\mathrm{n}}(v_i)$ represents the Min–Max normalized value of the hyperdegree $\mathrm{HDC}(v_i)$, while $\mathrm{DK}^{\mathrm{n}}(v_i)$ denotes the Min–Max normalized value of $\mathrm{DK}(v_i)$. Both of these metrics enhance the comparability of the data.

This method emphasizes that the importance of a node is not solely associated with its core position but also with the core positions of its neighbors and the numbers of hyperedges involved in neighbors with different values of k_s.

3.3. Example description

In order to demonstrate advantages of the LGK, we consider a simple example of hypergraphs in Fig. 3. Figure 3(a) contains 18 nodes and 15 hyperedges, while Fig. 3(b) includes 23 nodes and 15 hyperedges.

In Fig. 3(a), nodes 1 and 3 have equal k_s values 1, which implies that their levels of importance are indistinguishable. At this point, when applying the k_s^{d} index, it is noted that $k_s^{\mathrm{d}}(1) = \sqrt{26} > k_s^{\mathrm{d}}(3) = \sqrt{2}$, thereby indicating that node 1 is deemed more significant than node 3. For nodes 2 and 3, the equality $k_s^{\mathrm{d}}(2) = k_s^{\mathrm{d}}(3)$ suggests that the k_s^{d} index alone is insufficient for assessing their relative importance. In this context, the GI index can be further examined, revealing that

$\mathrm{GI}(2) = 2 < \mathrm{GI}(3) = 5$, which substantiates the assertion that node 3 holds greater significance than node 2. However, when considering the GI for nodes 3 and 5 in Fig. 3(b), the relationship $\mathrm{GI}(3) = 10 > \mathrm{GI}(5) = 8$ contradicts the established understanding that node 5 is more important than node 3. This inconsistency can be illustrated through the example of a paper collaboration network, where author 3 has co-authored only one paper with eight collaborators, predominantly from the 1-shell, whereas author 5 has participated in three papers with collaborators across the 1-shell, 2-shell, and 3-shell.

In the following analysis, we compute the LGK values for nodes 1, 2, 3, 4 and 5 in Fig. 3(b). The primary computations are the DK values as follows: $\mathrm{DK}(1) = 1 \times (1 \times 4 + 2 \times 2 + 3 \times 5) = 23$, $\mathrm{DK}(2) = 5$, $\mathrm{DK}(3) = 16$, $\mathrm{DK}(4) = 81$, $\mathrm{DK}(5) = 19$, $\mathrm{DK}_{\max} = 81$, and $\mathrm{DK}_{\min} = 2$. Then, the LGK values are $\mathrm{LGK}(1) = 1 + 21/79$, $\mathrm{LGK}(2) = 3/79$, $\mathrm{LGK}(3) = 14/79$, $\mathrm{LGK}(5) = 1/2 + 17/79$, and $\mathrm{LGK}(4) = 3/4 + 1$. As a result, the importance ranking of the nodes is $\mathrm{LGK}(4) > \mathrm{LGK}(1) > \mathrm{LGK}(5) > \mathrm{LGK}(3) > \mathrm{LGK}(2)$, which aligns with the observed reality.

3.4. Algorithm description and complexity analysis

The algorithm of the LGK is given in Algorithm 1. Specifically, lines 3–4 obtain the k_s value of all nodes by K-shell decomposition; lines 5–7 and 8–15 calculate the hyperdegrees of all nodes and DK values of all nodes by Eqs. (1) and (8), respectively; lines 16–20 calculate the LGK values of all nodes by Eq. (9).

Algorithm 1 LGK algorithm

1: **Input:** Hypergraph $H = (V, E)$
2: **Output:** Node influence ranking list: $\{\mathrm{LGK}(v_i) \mid v_i \in V\}$
3: Initialize $k_s(v_i) \leftarrow 0$ for all $v_i \in V$.
4: Compute k_s values using K-shell decomposition.
5: **for all** $v_i \in V$ **do**
6: Compute $\mathrm{HDC}(v_i)$.
7: **end for**
8: **for all** $v_i \in V$ **do**
9: Initialize sum $\leftarrow 0$;
10: **for** $j = 1$ to K **do**
11: Compute $N_{\mathrm{h}}^{j}(v_i)$.
12: sum $\leftarrow$ sum $+ j \times N_{\mathrm{h}}^{j}(v_i)$.
13: **end for**
14: Compute $\mathrm{DK}(v_i) \leftarrow k_s(v_i) \times$ sum.
15: **end for**
16: Normalize $\mathrm{HDC}(v_i)$ values to get $\mathrm{HDC}^{\mathrm{n}}$.
17: Normalize $\mathrm{DK}(v_i)$ values to get DK^{n}.
18: **for all** $v_i \in V$ **do**
19: Compute $\mathrm{LGK}(v_i) \leftarrow \mathrm{HDC}^{\mathrm{n}}(v_i) + \mathrm{DK}^{\mathrm{n}}(v_i)$.
20: **end for**
21: Sort the list $\{\mathrm{LGK}(v_i) \mid v_i \in V\}$ in descending order.
22: **return** The ranking list of nodes.

Consider a hypergraph $H = (V, E)$ with N nodes. The average hyperdegree is denoted as $\langle k \rangle$ and the average cardinality is represented as $\langle s \rangle$. The time complexity of the LGK method is obtained as follows: The complexity of computing k_s of all nodes in the hypergraph is $O(N \times \langle k \rangle)$. The complexity of computing DK is $O(N \times \langle k \rangle \times \langle s \rangle)$. The complexity of hyperdegree normalization and DK normalization is $O(N)$. Therefore, the time complexity of LGK is $O(N \times \langle k \rangle + N \times \langle k \rangle \times \langle s \rangle) \approx O(N \times \langle k \rangle \times \langle s \rangle)$.

Significantly, in most cases, $\langle k \rangle$ and $\langle s \rangle$ are always much smaller than N. As a result, the LGK method exhibits low time complexity, making it appropriate for application in large-scale hypergraphs.

4. Experiment

4.1. Dataset

The datasets utilized in this research include senate-committees,[47] cat-edge-madison-restaurant-reviews,[48] house-committees,[47] cat-edge-vegas-bars-reviews,[48] cat-edge-music-blues-reviews,[49] and cat-edge-algebra-questions.[48] These datasets are sourced from real world and are accessible at https://www.cs.cornell.edu/arb/data/. Hypergraphs have been constructed based on these datasets. The fundamental information of these hypergraphs is presented in Table 2.

Table 2. Some basic information of six hypergraphs. N, M, $\langle k \rangle$, $hd_{\max}$, and $hd_{\min}$ represent the number of nodes, the number of hyperedges, the average hyperdegree, the maximum hyperdegree, and the minimum hyperdegree, respectively.

Hypergraph	N	M	$\langle k \rangle$	$hd_{\max}$	$hd_{\min}$
Senate	282	315	19.1773	63	1
Restaurant	565	601	8.1434	59	1
House	1290	341	9.1806	44	1
Bar	1234	1198	9.6151	147	1
Music	1106	694	9.4928	127	1
Algebra	423	1268	19.5319	375	1

4.2. Correlation

The Pearson correlation coefficient serves as a quantitative indicator for assessing the linear relationship between two variables, with possible values spanning from -1 to 1. The strength of the association increases as the absolute value approaches 1. In this section, we utilize the Pearson correlation coefficient[50] to analyze the relationships among seven methods utilized for the identification of vital nodes.

The resulting correlation matrix is illustrated in Fig. 4. Among the hypergraphs analyzed, the results produced by k_s^{d}, LGK, and HGC show a strong correlation, while the outcomes generated by HDC, BC, and CC also show a robust correlation among themselves. However, the correlation between the first group of methods and the latter is comparatively weak. It is noteworthy that, despite the LGK and HGC methods focusing on different dimensions, specifically, the former on the complexity of hypergraph structure and the latter on node distance, there exists a significant correlation between them.

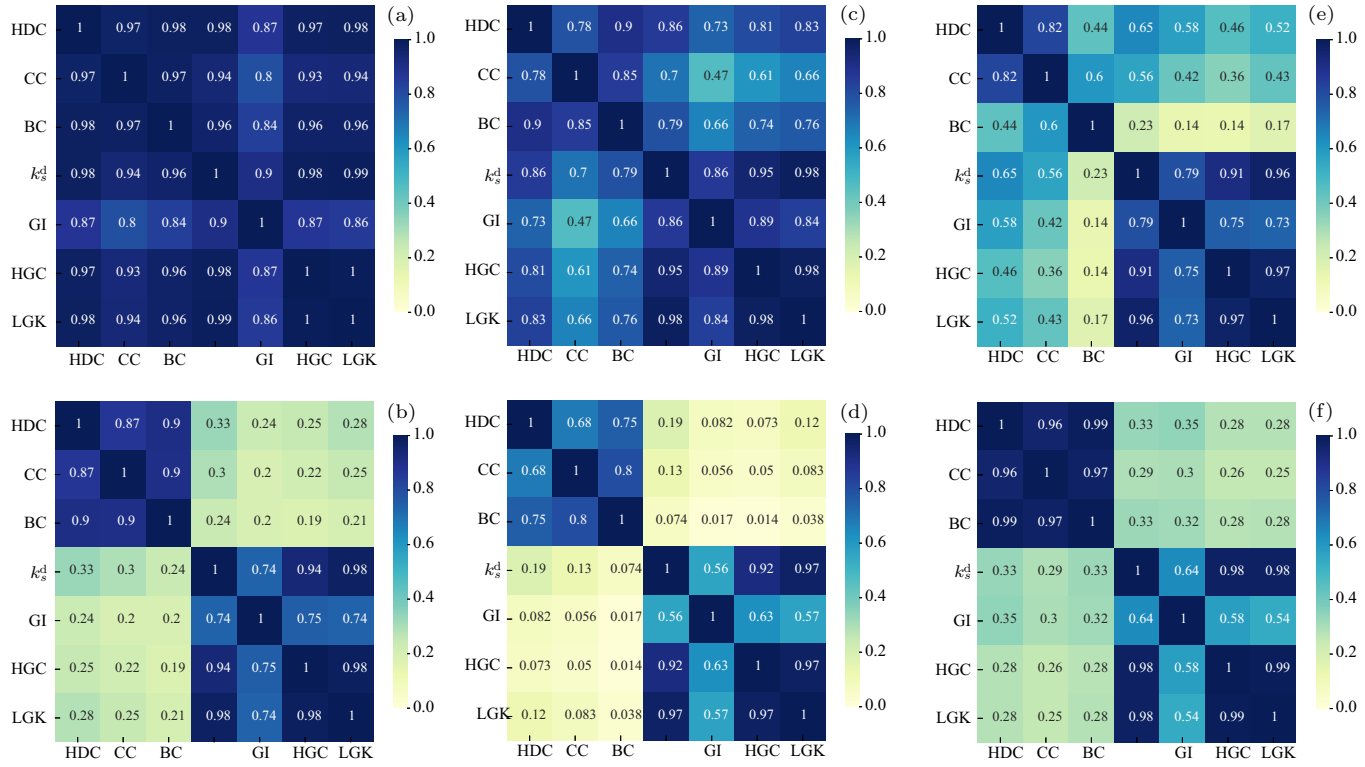


Fig. 4. The correlation matrices of seven methods in six hypergraphs. A deeper color signifies a higher Pearson correlation coefficient.

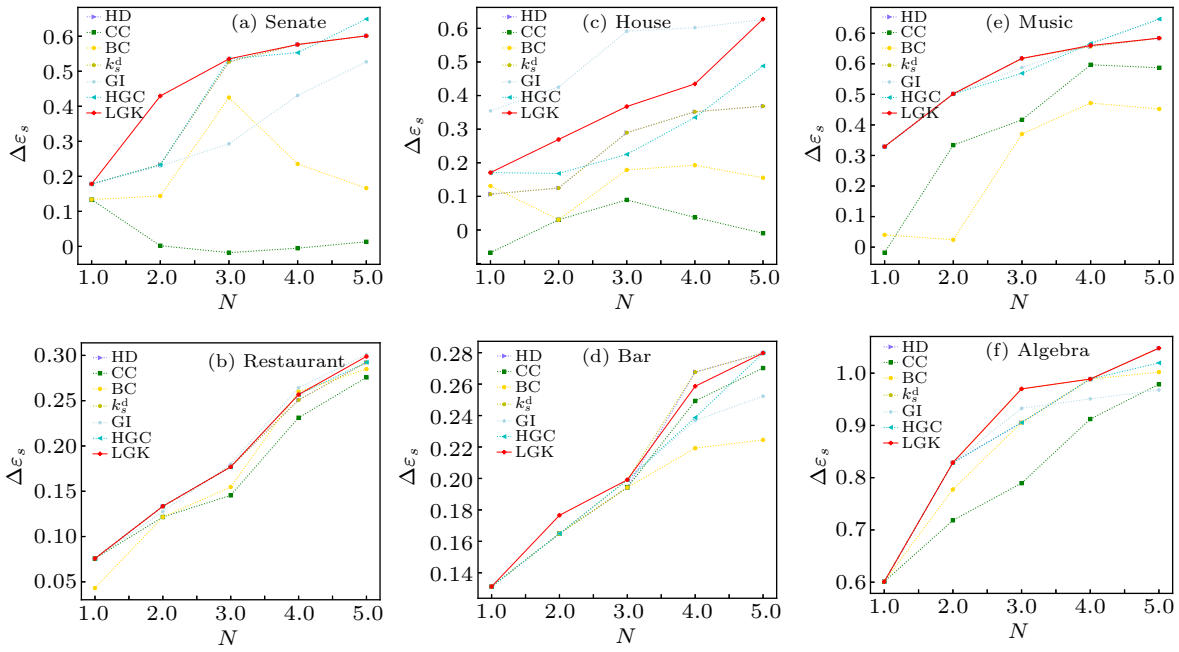


Fig. 5. Network loss efficiency of seven centrality metrics in six hypergraphs.

4.3. Network efficiency quantification

Network loss efficiency refers to the efficiency lost by a network following the removal of certain nodes along with their associated edges. The importance of removing nodes can be judged by the magnitude of network loss efficiency. The network loss efficiency of a hypergraph can be denoted as $\Delta\epsilon_s$.

In many real-world applications, one is often concerned only with the most influential, top-ranked nodes. To capture this focus, we designed a network loss efficiency experiment. Specifically, we treated the top 1, 2, 3, 4 and 5 nodes as initial failure seeds — removing each set of nodes together with their incident hyperedges — and then measured the resulting network loss efficiency.

Figure 5 presents comparative experiment results of seven methods in six distinct hypergraphs. The X-axis denotes the number of nodes initially removed, while the Y-axis measures

the resulting network loss efficiency — a higher value reflecting a greater deterioration of network integrity and, hence, the relative importance of the removed node subset. Overall, the LGK method demonstrates remarkably consistent performance across six different types of hypernetworks. In nearly all scenarios, the network efficiency loss caused by removing critical nodes continuously increases with the number of nodes removed, exhibiting a steady upward trend without significant fluctuations or declines. This result underscores the consistency, reliability, and robustness of the LGK method in identifying critical nodes. Comprehensively observing results from all networks, the LGK method consistently induces a higher network efficiency loss when removing a small number of top-ranked nodes. In most hypernetworks, LGK significantly outperforms traditional approaches such as HD, CC, BC, and k_s^{d}, and is comparable or superior to more computationally complex methods like GI and HGC. These findings indicate that the LGK method achieves higher precision and effectiveness in identifying the most critical nodes, highlighting its advantage of integrating structural characteristics from both the nodes themselves and their neighborhoods. Notably, when a few nodes (the top 1–3 nodes) are removed, LGK generally exhibits a pronounced advantage, further validating its broad applicability in practical scenarios.

4.4. Node spreading influence quantification

To assess the ability of the LGK method on node spreading influence, we perform numerical simulations of SIR spreading dynamics, utilizing the LGK method alongside six additional methods.

It is widely recognized that decision-makers tend to prioritize the highest-ranked nodes. Accordingly, the top 1/20 of nodes, as determined by each baseline method and the LGK method, are designated as the initial I-state nodes, while the remaining nodes are classified as the S-state nodes.

Figure 6 illustrates the transmission curves for the first 5 time steps with time t depicted on the horizontal axis and the total proportions of infected and recovered nodes shown on the vertical axis. The curves are differentiated by color to represent various methods for identifying vital nodes, with the red curve corresponding to the LGK method. It is evident that across all networks, the curves associated with the LGK method exhibit the steepest ascent and are positioned at the highest levels in the majority of hypergraphs. Consequently, the vital nodes identified by the LGK method have a more rapid and extensive influence on other nodes, thereby indicating their superior effectiveness in terms of spreading influence among nodes.

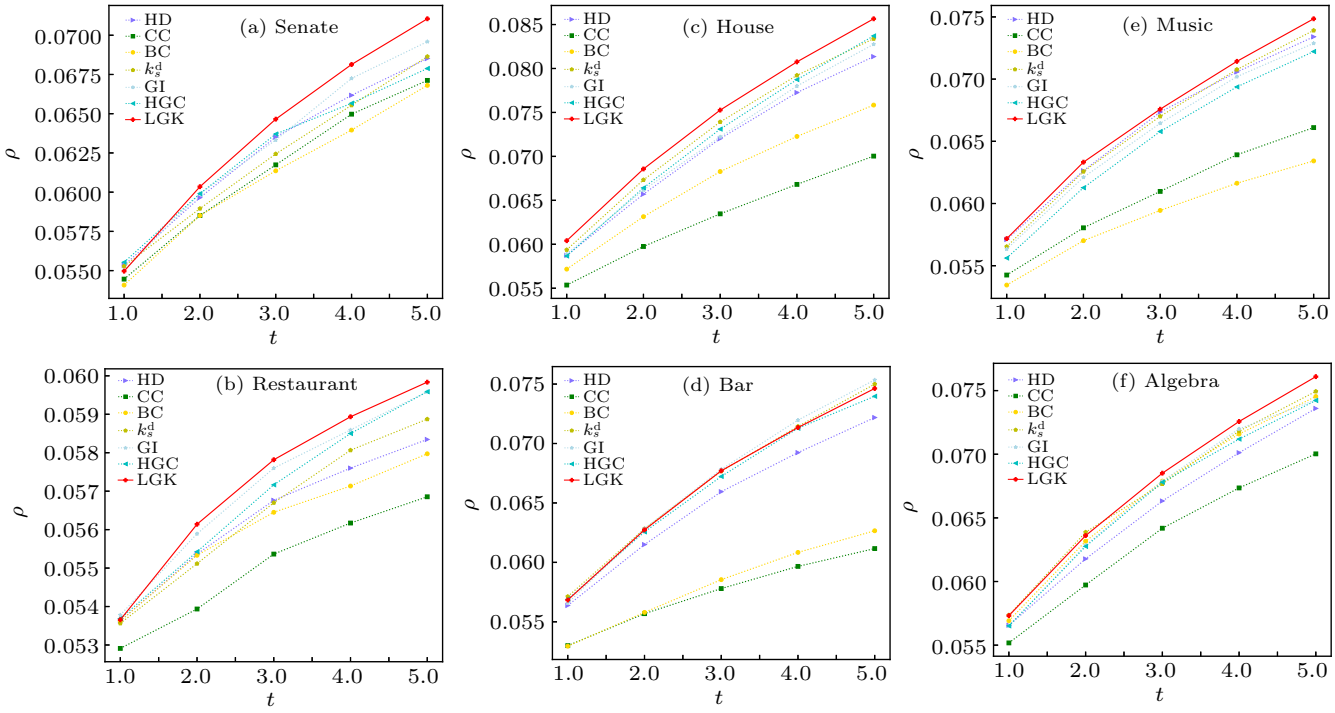


Fig. 6. Spreading curves of the top 1/20 ranked nodes according to seven methods in six hypergraphs.

4.5. Differentiation

In evaluating the performance of the LGK method, it is essential to employ an approach that not only accurately identifies nodes of significant influence but also effectively differentiates among all nodes. This differentiation is essential for a comprehensive understanding of the network's structure and dynamic characteristics. In the differentiation experiment, the effectiveness of a method is demonstrated by quantifying the number of nodes that receive identical scores from that method. Specifically, a high number of nodes with the same

score suggests a limited capacity to differentiate node influence.

The experimental results are illustrated in Fig. 7, where the red nodes indicate the ranking distribution obtained by the LGK method. As depicted in the figure, the majority of red nodes are positioned towards the lower end of the ranking, nearly forming a straight line. This observation indicates a low ranking overlap rate obtained by the LGK method, thereby suggesting that it possesses a strong ability to distinguish critical nodes in the network.

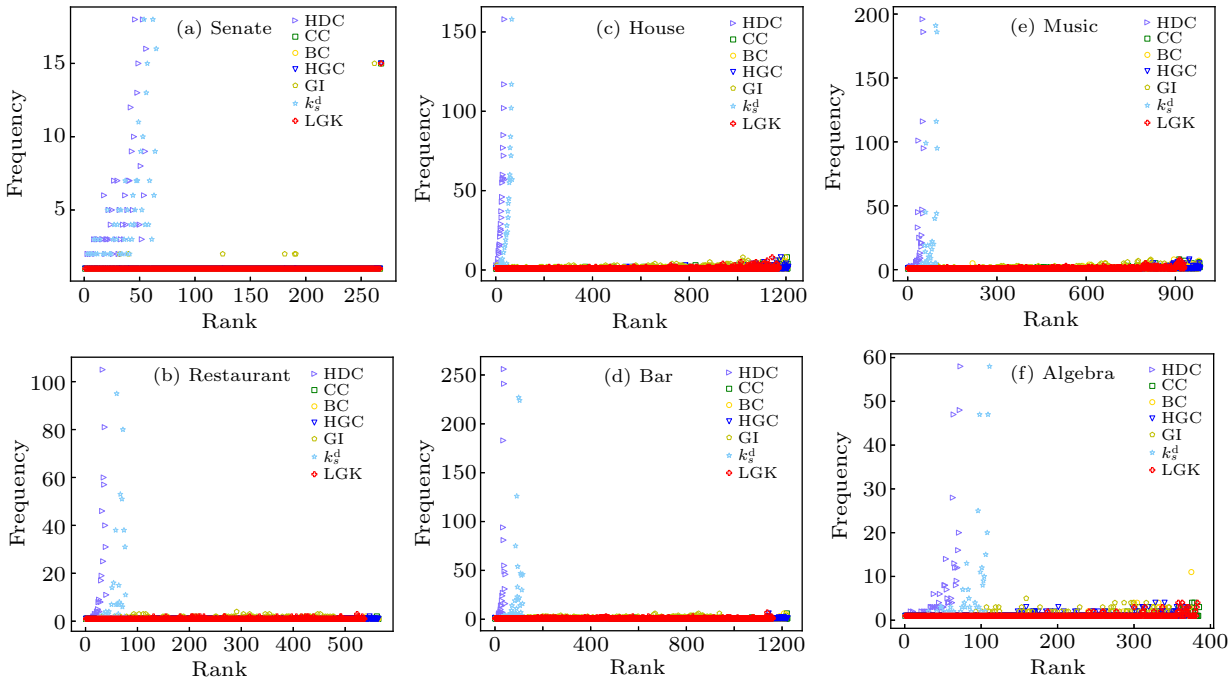


Fig. 7. Ranking frequency distribution obtained by seven methods in six hypergraphs.

Table 3. Comparison of the advantages and disadvantages of the seven methods.

Index	Perspective	Advantage	Disadvantage
HDC	local	Simple and straightforward, high computational efficiency	Fails to capture the relative positions of the nodes in global network structure
CC	global	Reflects reachability efficiency	High computational complexity, ineffective in networks containing multiple disconnected components
BC	global	Captures shortest paths, global influence	High computational complexity, sensitive to network changes
HGC	global	Incorporates degree and higher-order distance, more accurate	High computational complexity, unsuitable for large-scale networks
k_s^d	global + local	Incorporates degree and core position, suitable for large-scale networks	Fails to fully consider the detailed structural characteristics of the node neighborhood
GI	global	Considers the core positions of both the node and its neighbors	Fails to distinct the specific impacts of neighbors in different K-shell levels
LGK	global + local	Considers both the core position of nodes and the specific impacts of neighbors in different K-shell levels, achieves a balance between precision and time complexity	Compared with purely local indexs, the calculation is more complex

5. Conclusion and prospect

In this study, we introduce an innovative method for the identification of vital nodes in hypergraphs, termed the LGK method. This method is grounded in K-shell decomposition and effectively integrates both global and local perspectives, while also maintaining a relatively low time complexity.

We evaluate the efficacy of the LGK method across four dimensions: correlation, network loss efficiency, spreading influence, and differences, employing six real-world hypernetwork datasets for our analysis. The results obtained from the experiments suggest that, from the perspective of functional importance, the failure of top-ranked nodes identified by LGK will lead to a considerable increase in the network loss efficiency. Furthermore, from the perspective of dynamic properties, vital nodes identified by the LGK method exhibit a more significant spreading influence.

Consequently, the LGK method effectively reduces time complexity while maintaining a comparatively high level of accuracy, making it suitable for application in large-scale hypergraphs. However, existing studies[51] have shown that centrality measures based solely on network structure often fail to accurately predict the effectiveness of information spreading in real-world settings. This is because actual diffusion processes are significantly influenced by individual-level behavioral attributes, such as influence and susceptibility. Therefore, future research could focus on integrating individual behavioral characteristics with structural information to enhance the identification of influential nodes in hypergraphs.

Program availability

The source code is openly available in Science Data Bank at https://www.doi.org/10.57760/sciencedb.j00113.00247.

References

[1] Strogatz S 2001 *Nature* **410** 268
[2] Dorogovtsev S N, Goltsev A V and Mendes J F F 2008 *Rev. Mod. Phys.* **80** 1275
[3] Newman M E J 2003 *SIAM Review* **45** 167
[4] Cohen R, Havlin S and ben-Avraham D 2002 *Handbook of Graphs and Networks: From the Genome to the Internet* pp. 85–110
[5] Benson A R, Gleich D F and Leskovec J 2016 *Science* **353** 163
[6] Lawyer G 2015 *Scientific Reports* **5** 8665
[7] Pastor-Satorras R, Castellano C, Van Mieghem P and Vespignani A 2015 *Rev. Mod. Phys.* **87** 925
[8] Restrepo J G, Ott E and Hunt B R 2006 *Phys. Rev. Lett.* **97** 094102
[9] Liu J, Xiong Q Y, Shi W R, Shi X and Wang K 2016 *Physica A* **452** 209
[10] Hu P, Fan W L and Mei S W 2015 *Physica A* **429** 169
[11] Yang Y Y and Xie G 2016 *Information Processing & Management* **52** 911
[12] Shah D and Zaman T 2011 *IEEE Transactions on Information Theory* **57** 5163
[13] Bell D C, Atkinson J S and Carlson J W 1999 *Social Networks* **21** 1
[14] Radicchi F, Fortunato S, Markines B and Vespignani A 2009 *Phys. Rev. E* **80** 056103
[15] Albert R, Jeong H and Barabási A L 2000 *Nature* **406** 378
[16] Motter A E and Lai Y C 2002 *Phys. Rev.E* **66** 065102
[17] Cohen R, Havlin S and Ben-Avraham D 2003 *Phys. Rev. Lett.* **91** 247901
[18] Clement C B, Bierbaum M, O'Keeffe K P and Alemi A A 2019 arXiv:1905.00075
[19] Ruepp A, Waegele B, Lechner M, Brauner B, Dunger-Kaltenbach I, Fobo G, Frishman G, Montrone C and Mewes H W 2010 *Nucleic Acids Research* **38**(suppl_1) D497
[20] Mastrandrea R, Fournet J and Barrat A 2015 *PloS One* **10**9 e0136497
[21] Kapoor K, Sharma D and Srivastava J 2013 *2013 IEEE 2nd network science workshop (NSW)* 152
[22] Vogiatzis C, Veremyev A, Pasiliao E L and Pardalos P M 2015 *Optimization Letters* **9** 615
[23] Benson A R 2019 *SIAM Journal on Mathematics of Data Science* **1**2 293
[24] Kovalenko K, Romance M, Vasilyeva E, Aleja D, Criado R, Musatov D, Raigorodskii A M, Flores J, Samoylenko I and Alfaro-Bittner K 2022 *Chaos, Solitons & Fractals* **162** 112397
[25] Zeng Y J, Huang Y M, Ren X L and Lü L Y 2024 *Information Sciences* **679** 121067
[26] Piao Y H, Wang J Y and Li K Z 2025 *Chin. Phys. B* **34** 048902
[27] Hu F, Liu M, Zhao J and Lei L 2019 *Complex Systems and Complexity Science* **15** 31
[28] Estrada E and Rodríguez-Velázquez J A 2006 *Physica A* **364** 581
[29] Zhou L N, Li F X, Gong Y C and Hu F 2021 *Complex Systems and Complexity Science* **18** 15
[30] Wu Y H, Li M D and Hu F 2023 *Complex Systems and Complexity Science* **20** 40
[31] Xie X W, Zhan X X, Zhang Z K and Liu C 2023 *Chaos* **33**1
[32] Wang P, Ling G, Zhao P, Pan W Q and Ge M-F 2024 *Chaos, Solitons & Fractals* **188** 115503
[33] Guo L, Liu W, Zhang X and Hu F 2025 *International Conference on Computer Application and Information Security* **13562** 546
[34] Wang W, Nie Y Y, Li W Y, Lin T, Shang M S, Su S, Tang Y, Zhang Y C and Sun G Q 2024 *Physics Reports* **1056** 1
[35] Suo Q, Guo J L and Sen A Z 2018 *Physica A* **495** 475
[36] Landry N W and Restrepo J G 2020 *Physica A* **30** 10
[37] Ran Y J, Deng X M, Wang X M and Jia T 2020 *Chaos* **30** 8
[38] Nie Y Y, Zhong X N, Lin T and Wang W 2022 *Applied Mathematics and Computation* **432** 127380
[39] St-Onge G, Iacopini I, Latora V, Barrat A, Petri G, Allard A and Hébert-Dufresne L 2022 *Communications Physics* **5** 25
[40] Zhang Z Y, Mei X H, Jiang H J, Luo X P and Xia Y 2023 *Applied Mathematics and Computation* **446** 127887
[41] Cui S X, Liu F Z, Liang L D, Jardón-Kojakhmetov H and Cao M 2025 *Automatica* **177**
[42] Zhou Y, Liu Y P, Yuan L, Zhuo Y H, Xu K S, Wu J and Zheng M H 2025 *Chin. Phys. B* **34** 038704
[43] Berge C 1973 *A Survey of Combinatorial Theory* **20** 15
[44] Bretto A 2013 *Hypergraph Theory An introduction* (Cham: Springer) pp. 209–216
[45] Aksoy S G, Joslyn C, Marrero C O, Praggastis B and Purvine E 2020 *EPJ Data Science* **9**1 16
[46] Kitsak M, Gallos L K, Havlin S, Liljeros F, Muchnik L, Stanley H E and Makse H A 2010 *Nat. Phys.* **6** 888
[47] Chodrow P S, Veldt N and Benson A R 2021 *Science Advances* **7** eabh1303
[48] Amburg I, Veldt N and Benson A R 2020 arXiv:2006.05645
[49] Ni J, Li J and McAuley J 2019 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* 188
[50] Cohen I, Huang Y, Chen J D, Benesty J, Huang Y T and Cohen I 2009 *Noise reduction in speech processing* 1
[51] Zhou F, Lü L Y, Liu JG and Mariani M S 2024 *National Science Review* **11** nwae073