Information Processing and Management 60 (2023) 103161

Contents lists available at ScienceDirect

Information Processing and Management

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

An efficient adaptive degree-based heuristic algorithm for influence maximization in hypergraphs

Ming Xie [a], Xiu-Xiu Zhan [a,*], Chuang Liu [a,*], Zi-Ke Zhang [b,*]

[a] Research Center for Complexity Sciences, Hangzhou Normal University, Hangzhou, 311121, PR China
[b] College of Media and International Culture, Zhejiang University, Hangzhou 310058, PR China

ARTICLE INFO

Dataset link: https://github.com/DDMXIE/Influence-maximization-on-hypergraphs

Keywords:
Influence maximization
Hypergraphs
Spreading dynamics
Complex networks

ABSTRACT

Influence maximization (IM) has shown wide applicability in immense fields over the past decades. Previous researches on IM mainly focused on the dyadic relationship but lacked the consideration of higher-order relationship between entities, which has been constantly revealed in many real systems. An adaptive degree-based heuristic algorithm, i.e., *Hyper Adaptive Degree Pruning (HADP)* which aims to iteratively select nodes with low influence overlap as seeds, is proposed in this work to tackle the IM problem in hypergraphs. Furthermore, we extend algorithms from ordinary networks as baselines. Results on 8 empirical hypergraphs show that HADP surpasses the baselines in terms of both effectiveness and efficiency with a maximally 46.02% improvement. Moreover, we test the effectiveness of our algorithm on synthetic hypergraphs generated by different degree heterogeneity. It shows that the improvement of our algorithm effectiveness increases from 2.66% to 14.67% with the increase of degree heterogeneity, which indicates that HADP shows high performance especially in hypergraphs with high heterogeneity, which is ubiquitous in real-world systems.

1. Introduction

As a classical optimization problem, IM aims to identify K initial spreaders that maximize the influence spread under a certain spreading dynamics in a network. Due to its abundant applications, e.g., the control of disease (Cheng et al., 2020; Singh et al., 2021), the dissemination of information (Lei et al., 2015) and marketing management (Domingos & Richardson, 2001; Huang et al., 2019), the problem is widely studied in recent years. IM problem was first proposed to find the most helpful customers in viral marketing. Later on, Kempe et al. (2003) provided an approximation algorithm with provable guarantee, namely greedy, to target the influential seed nodes. In addition, the CELF method and its improved variant CELF++ were designed respectively (Leskovec et al., 2007). Moreover, there are many other methods designed to enhance the algorithm performance of IM (Gong et al., 2021; Li et al., 2021), including MIA (Chen et al., 2010), PMIA (Wang et al., 2012), etc.

Extensive researches of IM (Biswas et al., 2021; Wang et al., 2021) are oriented to ordinary networks, where edges were used to denote pairwise interactions between individuals. In many real-world scenarios, an edge in ordinary networks with dyadic relationship can hardly characterize the interactions if the interactions involve more than two entities. For example, multiple users may form groups for information sharing in social platforms, more than two researchers may contribute to one scientific paper, and many people might be listed in mass emails. This kind of relations can be represented by a hypergraph (Cencetti et al., 2021; Young et al., 2021) with hyperedges characterizing the polyadic interactions among more than two nodes (Ouvrard, 2020). In light of IM in hypergraphs, it is still a mostly unexplored problem with only a few studies focusing on this field. For instance, Zhu et al. (2018)

* Corresponding authors.
E-mail addresses: zhanxiuxiu@hznu.edu.cn (X.-X. Zhan), liuchuang@hznu.edu.cn (C. Liu), zkz@zju.edu.cn (Z.-K. Zhang).

https://doi.org/10.1016/j.ipm.2022.103161
Received 9 August 2022; Received in revised form 13 October 2022; Accepted 7 November 2022
Available online 24 November 2022

proved that IM in directed hypergraphs under IC model is an NP-hard problem, and they designed a sandwich framework with high computational complexity. In addition, a set of greedy-based heuristic strategies were proposed to address the minimum target set selection problem in hypergraphs (Antelmi et al., 2021). However, current researches either considered to transform hypergraphs to bipartite graphs or designed greedy algorithms to deal with the IM in hypergraphs, ignoring the basic hypergraph topological structures which may play a crucial role in tackling the IM. Even though the techniques above are applicable to some specific cases, the IM problem in a hypergraph still embraces several major challenges. The first one is to achieve a balance between effectiveness and efficiency. Algorithms that obtain optimal solution often takes a great deal of time, and thus the algorithms could hardly be applicable to hypergraphs with large size. Therefore, it is necessary to design efficient methodologies that could approximate the influence spread of the seed nodes to the optimal solution as much as possible. In addition, how to design a spreading dynamics that could evaluate the spreading influence of nodes in a hypergraph is still less explored in the previous studies. Last but not least, some network-based methods have already been proposed to tackle the IM problem in ordinary networks. However, how to extend the algorithms from ordinary network to hypergraph with considering the high-order topology is also a challenging issue. The goal of this work is to utilize the basic topological properties of a hypergraph to address IM problem in hypergraphs.

Degree centrality, as an essential topological property, was frequently used to characterize the node importance in a network (Lü et al., 2016; Stegehuis & Peron, 2021). In this study, we deal with the problem of how to choose the initial seeds for IM in hypergraphs based on the node degree. First, we investigate the hypergraphs generated by the real-world data and show the high influence overlap between nodes and their neighbors. Second, a discrete-time susceptible–infected (SI) model with Contact Process is designed to quantify the influence spread of seed nodes. Then, we propose the Hyper Adaptive Degree Pruning (HADP) algorithm for hypergraph IM, which iteratively avoids choosing nodes that have large influence overlap with the existing seeds as the seed candidates. Experiments indicate that HADP algorithm surpasses other baselines efficiently and accurately on both empirical and synthetic hypergraphs. Our main contributions are summarized as follows:

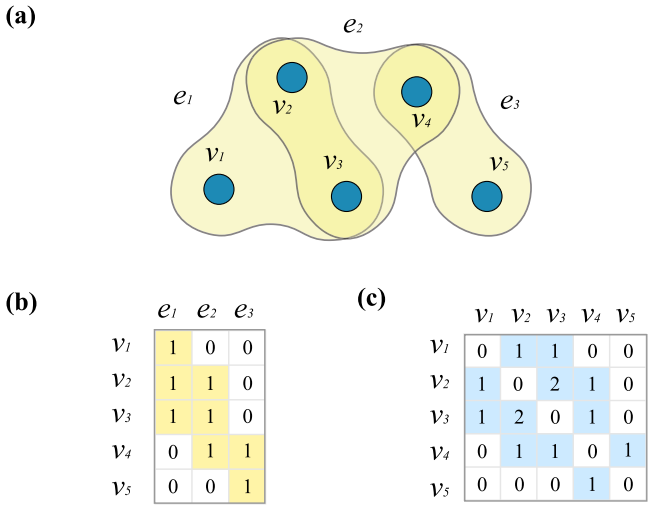
- We explore the IM problem in hypergraphs, and an adaptive degree-based heuristic algorithm named Hyper Adaptive Degree Pruning (HADP) is put forward to tackle the IM problem in hypergraphs under a discrete-time susceptible–infected (SI) spreading dynamics.
- Experimental results show that our algorithm outperforms the algorithms extended from the network-based methods and the state-of-the-art methods with both high effectiveness and efficiency.
- The performance of our method over the change of degree heterogeneity on synthetic hypergraphs is further explored. We find that our algorithm achieves better effectiveness in hypergraphs with higher degree heterogeneity.

We organized the remainder of the study as follows. To start with, Section 2 introduces the current researches that are related to our work. The preliminary definitions of a hypergraph and the problem statement are given in Section 3. Section 4 illustrates the spreading dynamics we used as well as the IM algorithms in hypergraphs. In Section 5, we provide detailed results and experimental analysis. We highlight the theoretical and practical implications and draw the conclusions in Sections 6 and 7.

2. Related works

IM problem (Qiu et al., 2021; Wang et al., 2022) was mostly based on the ordinary networks previously. The solutions for IM problem can be generally classified into the following categories, i.e., approximation algorithm, heuristic solutions and community-based approaches. In the study proposed by Kempe et al. (2003), an algorithm that guarantees the approximation rate of $(1-\frac{1}{e}-\epsilon)$ for selecting the seed nodes was presented, which was named as the greedy algorithm. However, the limitation of its filtering conditions leads to a high time cost. Hereafter, algorithms have been proposed to optimize the approximate solution to balance the effectiveness and efficiency. For example, CELF and CELF++ were proposed based on the rule of diminishing marginal gains (Goyal et al., 2011). Although these algorithms get some improvements in efficiency, the time complexity is rather high when performed on large-scale networks. Therefore, more efficient heuristic algorithms were proposed, such as the IRIE algorithm designed by Jung et al. (2012). IRIE combines the estimation of the influence spread of seeds and the influence ranking process to verify the algorithm effectiveness in the IC and its extension IC-N model. In addition, there are also some centrality-based heuristics that solve IM problem by selecting top-ranked nodes as seeds, such as degree and PageRank centrality (Brin & Page, 1998). Community-based methods were put forward to effectively diminish the influence overlap among seeds, such as C2IM (Singh et al., 2019), LKG (Samir et al., 2021) and INCIM (Bozorgi et al., 2016). Additionally, plenty of new algorithms investigated the IM problem from different perspectives have emerged in recent years, e.g., Li, Li et al. (2022) focused on the IM problem by considering the crowd emotion and Kumar et al. (2021) studied it in social networks based on label propagation model.

Higher-order interactions between entities have been continuously found on a variety of real systems, but only a few studies have focused on IM on higher-order networks. Amato et al. (2017) modeled the social media network via a hypergraph, in which user-to-multimedia relationships are represented by hyperedges. In their work, TIM+ and IMM were further applied to tackle the IM problem in a hypergraph after transforming it to a bipartite graph. A ranking-based algorithm was proposed under the HyperCascade model (Ma & Rajkumar, 2022), where the model considers spreading process on the bipartite augment graph of a hypergraph. The approach that is closely related to our study is proposed by Zhu et al. (2018). They modeled the social interactions through a directed weighted hypergraph. Based on the IC model, a D-SSA method which inspired by the RIS sampling process (Borgs et al., 2014) was designed to solve the general weighted social IM problem.



(b)

	e_1	e_2	e_3
v_1	1	0	0
v_2	1	1	0
v_3	1	1	0
v_4	0	1	1
v_5	0	0	1

(c)

	v_1	v_2	v_3	v_4	v_5
v_1	0	1	1	0	0
v_2	1	0	2	1	0
v_3	1	2	0	1	0
v_4	0	1	1	0	1
v_5	0	0	0	1	0

Fig. 1. An illustration example of (a) a hypergraph; (b) the incidence matrix of (a); (c) the adjacency matrix of (a).

3. Preliminary definition

3.1. Definition of a hypergraph

A hypergraph is represented as $H(V, E)$. $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$ stands for the node set and the hyperedge set, respectively. An incidence matrix of H is given by $C_{[n\times m]} = c_{i\alpha}$, where

$$c_{i\alpha} = \begin{cases} 1 & \text{if } v_i \in e_\alpha \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Therefore, the adjacency matrix $A_{[n\times n]}$ can be derived from C,

$$A_{ij} = [CC^T - D]_{ij}, \tag{2}$$

where D is a diagonal matrix, D_{ii} represent the number of hyperedges node i belongs to, and A_{ij} denotes the number of hyperedges which contain both node v_i and node v_j. An example of a hypergraph is given in Fig. 1, which contains 5 nodes and 3 hyperedges. The incidence matrix C and adjacency matrix A are also given correspondingly.

Given the incidence matrix of a hypergraph, the node degree and hyperdegree are further given as follows (Battiston et al., 2020). The degree of a node v_i ($deg(i)$) indicates the number of neighboring nodes of v_i, which is formally defined as:

$$deg(i) = \sum_{j=1}^{n} \tilde{A}_{ij}, \tag{3}$$

where $\tilde{A}$ is the binarized adjacency matrix of A, whose element $\tilde{A}_{ij} = 1$ if node v_i and node v_j share at least one hyperedge, and $\tilde{A}_{ij} = 0$ otherwise. In detail, it can be defined as follows:

$$\tilde{A}_{ij} = \begin{cases} 1 & if \quad A_{ij} > 0 \\ 0 & if \quad A_{ij} = 0 \end{cases} \tag{4}$$

The hyperdegree of node v_i is defined as the number of hyperedges to which node v_i belongs:

$$d^H(i) = \sum_{j=1}^{m} C_{ij} \tag{5}$$

According to the above definitions, we can calculate the degree and hyperdegree of the nodes in Fig. 1. For instance, the degree of node v_3 is $deg(3) = 3$ and the hyperdegree of node v_3 is $d^H(3) = 2$.

3.2. Problem statement

The study mainly addresses the problem of hypergraph influence maximization (HIM), which aims to identify K influential spreaders in a hypergraph under a specific spreading mechanism.

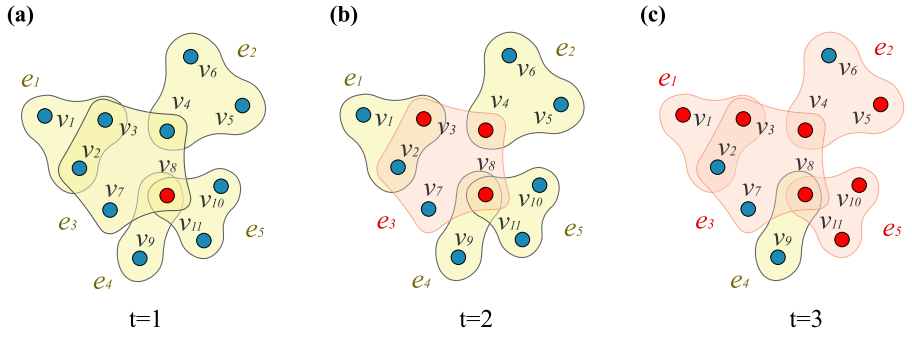


Fig. 2. An schematic diagram of the SI spreading dynamics with Contact Process.

The mathematical statement of the HIM problem is described as:

$$\begin{aligned} &\arg\max \{\sigma(S)\},\ S \subseteq V \\ &\quad \text{s.t. } |S| = K, \end{aligned} \tag{6}$$

where the number of nodes K in the seed set is the constraint condition of this problem, and $\sigma(S)$ is the expected influence of the seed node set S ($S \subseteq V$).

IM problem in ordinary networks has been proved to be NP-hard in Kempe et al. (2003). The HIM problem, which can be considered as the generalization of IM in ordinary networks, is also NP-hard (Zhu et al., 2018). That is to say, it cannot be solved in polynomial time. As a result, we propose to use heuristic algorithms and greedy algorithms to approximate its optimal solution.

4. Algorithms

4.1. Susceptible–infected spreading model with Contact Process dynamics

To quantify the spreading influence of the seed nodes (Ferraz de Arruda et al., 2021; Zhan et al., 2020), we propose to use a Susceptible–Infected (SI) model with Contact Process (CP) dynamics on a hypergraph (Suo et al., 2018). In the model, an individual can only in either susceptible (S) or infected (I) state. An S-state node can be infected by each of its neighbors in I-state with an infection rate β. The SI model in hypergraphs is described as follows:

- **Step 1**: Initially, nodes in the seed set are set to be infected, and the rest nodes are in susceptible.
- **Step 2**: At each time step t, we first find the I-state nodes. For each I-state node v_i, we find all the hyperedges $E_i = \{e_{i1}, e_{i2}, \ldots, e_{iq}\}$ that node v_i belongs to. Then a hyperedge e is chosen from E_i uniformly at random. For each of the S-state nodes in e, it will be infected by node v_i with infection probability β.
- **Step 3**: We terminate the process until a specific time step T reaches, where T is a control parameter.

We show an illustrative instance of SI spreading process in hypergraphs in Fig. 2. At time step $t = 1$, node v_8 is in I-state. The hyperedge set that contains v_8 is $E_8 = \{e_3, e_4, e_5\}$. At time step $t = 2$, the S-state nodes, i.e., v_3 and v_4 in hyperedge e_3, are infected by node v_8. Subsequently, the I-state nodes v_3, v_4 and v_8 infect the S-state nodes in hyperedges e_1, e_2, e_5.

4.2. Adaptive degree-based heuristic algorithms

Given nodes v_i and v_j, we suppose that the influenced node sets at time step T by setting node v_i and v_j as the seed node are given by $I_T(v_i)$ and $I_T(v_j)$, respectively. Thus, the influence overlap o_{ij}^T at time step T between v_i and v_j can be defined as $o_{ij}^T = \frac{I_T(v_i) \cap I_T(v_j)}{n}$. In Fig. 3, we show the comparison between the influence overlap distribution of a neighboring node pair as well as a randomly selected node pair in various hypergraphs. A detailed description will be given in Section 5.1. In most datasets (i.e., Fig. 3(a), (b), (e), (g) and (h)), the probability that a neighboring node pair have overlapped influence is always higher than that of a randomly selected node pair. It suggests that when we choose one node as the seed, the probability that its neighboring nodes are choosing as the seed should be diminished to avoid overlapped influence. Based on this assumption, we propose an adaptive degree-based heuristic algorithm, i.e., Hyper Adaptive Degree Pruning (HADP), to solve the HIM problem.

Hyper Adaptive Degree Pruning (HADP). In HADP, we aim to punish nodes that have more neighbors in S in each iteration. The details are given in Algorithm 1. To conduct the HADP, we first give the original degree vector of all the nodes as $deg^0 = (deg^0(1), deg^0(2), \ldots, deg^0(n))$.

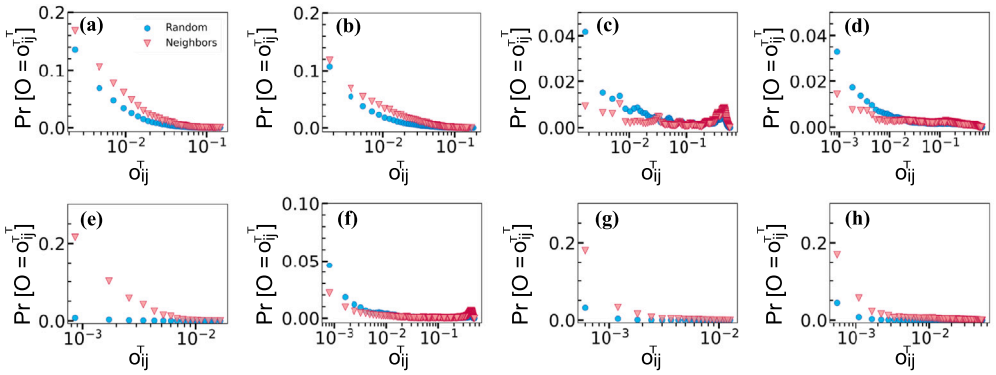


Fig. 3. The influence overlap distribution of a randomly selected node pair (blue) and a neighboring node pair (pink) in dataset (a) Algebra; (b) Restaurant-Rev; (c) Geometry; (d) Music-Rev; (e) NDC-classes; (f) Bars-Rev; (g) iAF1260b; (h) iJO1366. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- **Step 1**: At the initial step, a node v_i that has the largest degree is added to the seed set S, i.e., $deg^0(i) = \max\{deg^0\}$. For every neighboring node v_u of v_i (i.e., $v_u \in N(i)$), For every neighboring node v_u of v_i, we first find the neighbors of v_u in S and collect all of them as a adaptive set $N_S(u)$. Then the adaptive degree of node v_u is updated as $deg^1(u) = deg^0(u) - Z$, in which $Z = |N_S(u)|$ denotes the size of elements in $N_S(u)$. For the other nodes that are not the neighbors of v_i, e.g., node v_w, $deg^1(w) = deg^0(w)$. After updating the adaptive degree of every node, we obtain an adaptive degree vector $deg^1 = (deg^1(1), deg^1(2), \ldots, deg^1(n))$. In consideration of the overlap between hyperedges, the degree penalty of each node varies depending on the hyperedges that the node belongs to. Thus, we note that the elements in $N_S(u)$ can be duplicated.
- **Step 2**: At step k, the node not in S that has the largest adaptive degree (denoted as v_j ($v_j \in V \backslash S$)), i.e., $deg^{k-1}(j) = \max\{deg^{k-1}\}$, is chosen, and we add it to the seed set S. For every neighboring node v_q of v_j, we first find the adaptive neighbors of v_q in S and collect them as a set $N_S(q)$. The adaptive degree of v_q is further updated by $deg^k(q) = deg^{k-1}(q) - Z$, where $Z = |N_S(q)|$. For the other nodes that are not the neighbors of v_j, e.g., node v_w, $deg^k(w) = deg^{k-1}(w)$. We obtain a new adaptive degree vector as $deg^k = (deg^k(1), deg^k(2), \ldots, deg^k(n))$ after updating the adaptive degree of every node.
- **Step 3**: The algorithm is terminated when we obtain K seed nodes.

We propose a simplified algorithm which considers to give an even penalty for every node in the iterations, i.e., at step k, the adaptive degree of v_q is further updated as $deg^k(q) = deg^{k-1}(q) - Z$, where $Z = 1$. In this case, it should be noted that v_q is a node in set $N_S(j)$ whose elements can be repeated. This simplified algorithm is named as Hyper Single Degree Pruning (HSDP), which we will use as a baseline in the following sections.

Algorithm 1: Hyper Adaptive Degree Pruning (HADP)

Input : Size of seed nodes K
Hypergraph $H(V, E)$
Output: Seed node set S
Initialization: $deg^0 \leftarrow$ Degree of each node.
while $|S| \leq K$ **do**
  $k \leftarrow |S|$
  $v_j\ (v_j \in V \backslash S) \leftarrow max\{deg^k(j)\}$
  $S \leftarrow S \cup \{v_j\}$
  $N(j) \leftarrow$ Neighbors of node v_j
  for v_q *in* $N(j)$ **do**
    $N_S(q) \leftarrow$ Adaptive neighbor set of v_q in S
    Adaptive degree $Z \leftarrow |N_S(q)|$
    $deg^k(q) = deg^{k-1}(q) - Z$
  end
  for v_w *not in* $N(j)$ **do**
    $deg^k(w) = deg^{k-1}(w)$
  end
end

Table 1
Topological properties of the datasets. n and m represent the number of nodes and hyperedges in a hypergraph, respectively, $\langle deg \rangle$ is the average of node degree, $\langle d^H \rangle$ is the average of node hyperdegree, $\langle d^E \rangle$ represents the average of the size of the hyperedges, which is given by the number of nodes in the hyperedge. c, $\langle d \rangle$, ξ and ρ are the clustering coefficient, the average of shortest path length, diameter and edge density of the corresponding ordinary network of a hypergraph.

Hypergraphs	n	m	$\langle deg \rangle$	$\langle d^H \rangle$	$\langle d^E \rangle$	c	$\langle d \rangle$	ξ	ρ
Algebra	423	1268	78.90	19.53	6.52	0.79	1.95	5	0.19
Restaurant-Rev	565	601	79.75	8.14	7.66	0.54	1.98	5	0.14
Geometry	580	1193	164.79	21.53	10.47	0.82	1.75	4	0.28
Music-Rev	1106	694	167.87	9.49	15.13	0.62	1.99	8	0.15
NDC-classes	1161	1088	10.71	5.55	5.92	0.61	3.50	9	0.01
Bars-Rev	1234	1194	174.30	9.61	9.93	0.58	2.10	6	0.14
iAF1260b	1668	2351	13.26	5.46	3.87	0.55	2.67	7	0.007
iJO1366	1805	2546	16.91	5.55	3.94	0.58	2.62	7	0.009

5. Experiments

By utilizing the eight hypergraphs generated by real-world data, extensive experiments are conducted to verify the algorithm effectiveness and efficiency. Besides, the robustness of our algorithm was tested in synthetic hypergraphs generated by different degree heterogeneities as well. All the algorithms are written in Python and each of them runs on a Linux server with 2.20 GHz Intel(R) Xeon(R) Silver 4114 CPU and 90G memory.

5.1. Data description

We show the basic description and properties of eight hypergraphs generated by real-world datasets, which are collected from different domains.[1,2] The hypergraphs will be utilized to validate the algorithm performance in the subsequent sections. The topological properties of them are given in Table 1. The detailed description of each data is given as follows:

cat-edge-algebra-questions dataset (Algebra) & cat-edge-geometry-questions dataset (Geometry). The two datasets contain interactions between users on a mathematics website, i.e., MathOverflow. The interactions between users are mainly about comments, questions and answers on algebra (or geometry) problems. Each node represents a user on MathOverflow. Users who answered the same type of question (in the area of algebra or geometry) is represented by a hyperedge.

cat-edge-madison-restaurant-reviews (Restaurant-Rev). The data indicates users who reviewed a specific type of restaurants on Yelp within a month's time. Each node and each hyperedge represent a user on this website and the set of users who reviewed a certain restaurant, respectively.

cat-edge-music-blues-reviews (Music-Rev). The data contains nodes and hyperedges which separately represent the users on Amazon and the reviewers sets who reviewed a particular category of blues music within a month time frame.

cat-edge-vegas-bars-reviews (Bars-Rev). Each node in the dataset denotes a user on Yelp, and a hyperedge is a set of users who reviewed a certain bar in Las Vegas, NV.

NDC-classes. The dataset contains nodes representing class labels, and a hyperedge is a drug which consists of a set of class labels.

iAF1260b. The data contains nodes representing reaction-based metabolics, and hyperedges are sets of metabolics which are applied to a certain reaction. The duplicate hyperedges are removed.

iJO1366. Similar to **iAF1260b**, this is also a metabolic hypergraph with each node representing a reaction-based metabolic, and hyperedges are sets of metabolics which are applied to a certain reaction. The duplicate hyperedges are removed.

5.2. Extended algorithms and baselines

To verify the performance of our algorithm, we propose two algorithms extended from ordinary network, i.e., H-RIS and H-CI, and choose four other up-to-date algorithms, i.e., Greedy, HyperIMRANK, HyperDegree and Degree, proposed by other researchers as baselines. The details of each algorithm are given as follows.

Hyper Reverse Influence Sampling (H-RIS). Reverse Influence Sampling (RIS) algorithm was designed to tackle the IM problem in an ordinary network (Borgs et al., 2014). In this work, we extend the RIS to hypergraphs by first introducing the following two definitions:

Definition 1 (*Hyper Reverse Reachable Set*)**.** Given a hypergraph $H(V, E)$, we remove each hyperedge with probability $1 - \beta$ and obtain a sub-hypergraph $H'(V', E')$. Given a node $v \in V$, we define the hyper reverse reachable (HRR) node set as a collection that can reach node v in H'.

[1] https://www.cs.cornell.edu/~arb/data/.
[2] http://bigg.ucsd.edu/.

Definition 2 (*Random HRR Set*)**.** For a randomly selected node $v \in H$, a random HRR set is defined as a HRR set which is randomly sampled from the pruned hypergraph H'.

We illustrate the H-RIS algorithm in Algorithm 2, which mainly contains the following two steps:

- **Step 1**: We generate η random HRR sets, in which η is a tunable parameter.
- **Step 2**: In each round of seed selection, we add node v_q with the highest frequency in the generated HRR sets to the seed set S. Then, the HRR sets that contain node v_q are removed. The selection rounds is terminated until seed set contains K nodes.

Algorithm 2: Hyper Reverse Influence Sampling (H-RIS)

Input : Size of seed nodes K
Infection probability β
Hypergraph $H(V, E)$
Output: Seed node set S
Initialization: $S = \emptyset$, $U = \emptyset$. U is a set of HRR.
$H'(V', E') \leftarrow$ Remove hyperedges with probability $1 - \beta$ from hypergraph $H(V, E)$
for $i = 1$ *to* η **do**
  $v_i \leftarrow$ Pick out a node at random
  $HRR \leftarrow$ Aquire nodes reachable to v_i from $H'(V', E')$
  $U \leftarrow U \cup \{HRR\}$
end
while $|S| \leq K$ **do**
  $v_q \leftarrow$ Node with the highest frequency in U
  $S \leftarrow S \cup \{v_q\}$
  Delete the HRR containing v_q from U
end

The algorithm suggests that if a node appears more frequently in different HRR sets, it will have a higher probability to influence the other nodes. Correspondingly, the more HRR sets that the seed set S covers, the more likely that S will have a large expected influence. We set $\eta = 200$ to conduct the experiments.

Hyper Collective Influence (H-CI). Collective Influence (CI) was first proposed to select seed nodes by utilizing the degree of distant nodes in an ordinary network (Morone & Makse, 2015). We extend the algorithm to a hypergraph by substitute the degree with the hyperdegree and give the definition of hyper collective influence. A ball $Ball(v_i, l)$ is a node set whose elements contain all nodes within a ball whose radius is l, where l denotes the shortest path from a node in $Ball(v_i, l)$ to node v_i. The frontier of $Ball(v_i, l)$ is denoted as $\partial Ball(v_i, l)$, i.e., the path length of any node inside $\partial Ball(v_i, l)$ to node v_i equals to l. We define the HCI of node v_i, which is read as:

$$HCI_l(i) = (d^H(i) - 1) \sum_{v_j \in \partial Ball(v_i, l)} (d^H(j) - 1), \tag{7}$$

where $d^H(i)$ is the hyperdegree of node v_i.

Given a specific value of l, we compute the HCI of each node in the hypergraph and choose the top K nodes whose HCI value is the largest to be the seeds for HIM problem. In our work, the tunable parameter l is set as 1 and 2, and we name the algorithms as H-CI($l = 1$) and H-CI($l = 2$), respectively.

Greedy. Greedy algorithm gives a guaranteed approximation of influence spread by accurately approximating influence spread with high computational complexity. The algorithm can be extended to a hypergraph (Kempe et al., 2003), which is shown in Algorithm 3. We denote S_{k-1} as the seed nodes that are selected at round $k - 1$, the expected influence spread by S_{k-1} is given by $\sigma(S_{k-1})$. The marginal gain of influence spread at round k is given by $\sigma(S_{k-1} \cup \{v\}) - \sigma(S_{k-1})$. At the beginning of the algorithm, S is set to be empty. At round k, we calculate the expected influence spread $\sigma(S_{k-1} \cup \{v\})$ for each v, where $v \in V \backslash S_{k-1}$. Node v_k with the largest marginal influence contribution ($v_k = \arg\max_{v \notin S_{k-1}} \{\sigma(S_{k-1} \cup \{v\}) - \sigma(S_{k-1})\}$) is inserted into the seed set, i.e., $S_k = S_{k-1} \cup \{v_k\}$. The algorithm is terminated until the seeds set contains K nodes.

Hyper-IMRANK (H-IMRANK). As a generalized algorithm of IMRANK, H-IMRANK (Ma & Rajkumar, 2022) still aims at improving a node ranking by iteratively approximate the marginal influence of nodes and finally obtaining a convergent and self-consistent node sequence. The algorithm is performed under a HyperCascade spreading model with given influence probabilities, i.e., p_1 and p_2, in an augmented bipartite network of the original hypergraph. In each iteration, the marginal influence of nodes is estimated by a generalized strategy named HyperRank Last to First Allocating, and then the ranking is reorganized in a decreasing order based on the estimated values. The iterations of influence estimation and node re-ranking process are repeated until the node ranking converges. We set p_1 and p_2 as 0.01 and conduct the experiments. The top K nodes in the convergent ranking of the output are collected.

Algorithm 3: Greedy

Input : Size of seed nodes K
Hypergraph $H(V, E)$
Output: Seed node set S
Initialization: $S_0 = \emptyset$, $k = 1$.
while $|S| \leq K$ **do**
  $v_k = \arg\max_{v \notin S_{k-1}} \{\sigma(S_{k-1} \cup \{v\}) - \sigma(S_{k-1})\}$
  $S_k = S_{k-1} \cup \{v_k\}$
  $k = k + 1$
end

HyperDegree (H-Degree). We compute the hyperdegree of each node, i.e., d^H, in a hypergraph and arrange them in descending order. The seed nodes for IM problem consists of the top-ranked K nodes with the node hyperdegree.

Degree. Similar to HyperDegree, we compute the degree of each node in a hypergraph and sort them in descending order. The top-ranked K nodes are collected to be the seeds set.

5.3. Experimental evaluation on real-world data

To validate the algorithm performance, we use the seed set obtained by each algorithm as the seed nodes for the SI spreading model with contact process running on various hypergraphs. In the SI spreading model, we show the results of different combinations of infection probability β and the termination step. The value of $\sigma(S)$ is given by the average of the outbreak sizes over 500 realizations for each algorithm. In addition, the seed set size varies from 1 to 25 in our experiments.

The influence spread of the seed set selected by different algorithms when $\beta = 0.01, T = 25$ are given in Fig. 4 and Table 2. In Fig. 4, we depict the expected influence spread as a function of the seed set size K, and the normalized area under each of the influence spread curve (AUC) is further given in Table 2. The best performance is obtained by Greedy algorithm, which comprehensively considers the topological and dynamical information. The algorithms (i.e., HADP and HSDP) we proposed perform the second best in almost all the hypergraphs, except for hypergraph **Bars-Rev** with AUC slightly lower than H-RIS (i.e., 0.0008 lower than H-RIS). In particular, we find that HADP has maximally 46.02% improvement in effectiveness compared to other benchmarks from Table 2. As it is illustrated in Section 4.2, the basic assumption for HADP and HSDP is that when we choose one node as the seed, the probability that its neighboring nodes are choosing as the seed should be diminished to avoid overlapped influence. HADP, HSDP and Degree are algorithms based on the node degree, but HADP, HSDP perform much better than Degree algorithm in all the hypergraphs. In hypergraphs such as **Algebra**, **Restaurant-Rev**, **NDC-classes**, **iAF1260b** and **iJO1366**, the probability that a neighboring node pair have overlapped influence is higher than that of a randomly selected node pair (Fig. 3). Accordingly, the AUC values in these hypergraphs derived from HADP, HSDP are also relatively larger than other algorithms except Greedy, which is shown in Table 2. It suggests that the assumption of reducing influence overlap can help to refine the performance of HIM algorithms. The fact that HADP is superior to HSDP in finding seed nodes further implies that considering an uneven penalty for each node in the design of the algorithm is more reasonable for HIM. H-CI($l = 1$), H-CI($l = 2$) and H-Degree are algorithms based on the hyperdegrees of the nodes, and we find that H-CI($l = 1$) and H-CI($l = 2$) perform slightly better than H-Degree. It indicates that considering the hyperdegree of distant nodes can help to improve the selection of seeding nodes. The AUC values of the other combinations of β and T are given in Tables 3, 4, 5, respectively, which are consistent with those we obtained from $\beta = 0.01, T = 25$. In particular, the fact that the maximal effectiveness improvement of HADP reaches 43.64%, 45.72%, 44.48% demonstrates HADP is robust over different settings of spreading parameters in the spreading model.

We further show the time cost for singling out seed node set ($\beta = 0.01, T = 25, K = 25$) in Table 6, where the time cost is the average over 10 realizations for each algorithm. Even though Greedy algorithm performs the best for influence spread, it has the highest time cost, i.e., it takes hours or days for each realization. Besides, H-RIS and HCI($l = 2$) also have high computational complexity compared to the remaining algorithms. H-Degree and Degree take the least time cost but with low AUC. In contrast, HADP and HSDP can achieve relatively high AUC with low time cost (within 50 s) in all the hypergraphs.

5.4. Experimental evaluation on synthetic hypergraphs

The HIM methods we proposed, i.e., HADP and HSDP, are adaptive degree-based heuristic methods. To check the robustness of our method over the change of the degree heterogeneity (St-Onge et al., 2022), the performance of our methods on synthetic hypergraphs is evaluated. As the random hypergraph generator (i.e., HyperCL) we choose can only generate synthetic hypergraphs with given hyperdegree distribution (Lee et al., 2021), we first investigate the correlation between node degree and hyperdegree in real-world datasets. Fig. 5 indicates that the node degree is positively correlated with the corresponding hyperdegree in the hypergraphs generated by real data, with the Pearson Correlation Coefficient (PCC) higher than 0.5. It means that the hypergraph generator can also generate hypergraphs with various degree heterogeneities. The details of HyperCL are given as follows:

Initially, we suppose the hyperdegree and the hyperedge size sequence of a hypergraph $H(V, E)$ are given as $\{d^H(1), d^H(2), \ldots, d^H(n)\}$ and $\{d^E(1), d^E(2), \ldots, d^E(m)\}$, respectively. For each $e_i \in E$, the nodes belong to e_i are sampled independently. That is to

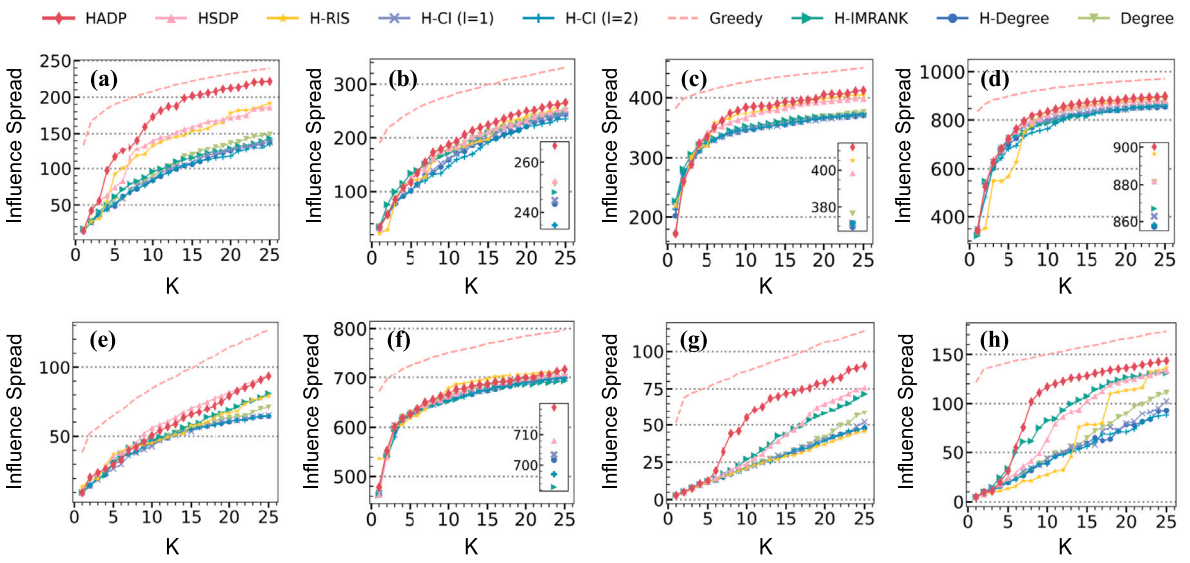


Fig. 4. Expected influence spread under different seed set size K for each algorithm in hypergraphs: (a) Algebra; (b) Restaurant-Rev; (c) Geometry; (d) Music-Rev; (e) NDC-classes; (f) Bars-Rev; (g) iAF1260b; (h) iJO1366. The subplots show the rankings of influence spread of seed nodes filtered by different algorithms in those data with small difference between algorithms when seed set size $K = 25$. We set $\beta = 0.01, T = 25$.

Table 2
AUC scores obtained by each of the curves shown in Fig. 4 for algorithms, i.e., HADP, HSDP, H-RIS, H-CI ($l = 1$), H-CI ($l = 2$), H-Degree and Degree. The best performance, i.e., the largest AUC score, is shown by $**$ and the second best is shown by $*$ in each hypergraph. We set $\beta = 0.01, T = 25$.

Hypergraphs	Algorithms (AUC)							
	HADP	HSDP	H-RIS	H-CI ($l = 1$)	H-CI ($l = 2$)	H-IMRANK	H-Degree	Degree
Algebra	**0.1844****	0.1487*	0.1453	0.1030	0.0994	0.1087	0.1009	0.1098
Restaurant-Rev	**0.1351****	0.1286*	0.1267	0.1207	0.1147	0.1277	0.1182	0.1283
Geometry	**0.1316****	0.1287*	0.1304	0.1215	0.1215	0.1232	0.1214	0.1217
Music-Rev	**0.1292****	0.1274*	0.1234	0.1248	0.1215	0.1234	0.1233	0.1270
NDC-classes	0.1399*	**0.1476****	0.1256	0.1128	0.1145	0.1266	0.1143	0.1187
Bars-Rev	0.1261	0.1255*	**0.1269****	0.1245	0.1239	0.1237	0.1241	0.1253
iAF1260b	**0.2110****	0.1445*	0.0959	0.1019	0.1000	0.1395	0.1002	0.1070
iJO1366	**0.1902****	0.1468	0.1139	0.0977	0.0909	0.1617*	0.0929	0.1058

Table 3
AUC scores obtained by our algorithms and baselines. The best performance, i.e., the largest AUC score, is shown by $**$ and the second best is shown by $*$ in each hypergraph. We set $\beta = 0.005, T = 35$.

Hypergraphs	Algorithms (AUC)							
	HADP	HSDP	H-RIS	H-CI ($l = 1$)	H-CI ($l = 2$)	H-IMRANK	H-Degree	Degree
Algebra	**0.2096****	0.1528*	0.1498	0.0958	0.0918	0.1026	0.0941	0.1038
Restaurant-Rev	**0.1401****	0.1288	0.1333*	0.1185	0.1094	0.1277	0.1139	0.1283
Geometry	**0.1384****	0.1323	0.1329*	0.1186	0.1186	0.1221	0.1180	0.1191
Music-Rev	0.1373*	0.1303	**0.1380****	0.1194	0.1130	0.1165	0.1163	0.1292
NDC-classes	0.1346*	**0.1456****	0.1144	0.1166	0.1180	0.1302	0.1182	0.1225
Bars-Rev	**0.1285****	0.1284*	0.1098	0.1269	0.1255	0.1266	0.1261	0.1281
iAF1260b	**0.1998****	0.1391*	0.1130	0.1031	0.1019	0.1335	0.1023	0.1072
iJO1366	**0.2230****	0.1563	0.0761	0.0927	0.0880	0.1727*	0.0883	0.1028

say, each node v_j selected into e_i is added in probability proportion (i.e., $\frac{d^H(j)}{\sum_{j=1}^{n} d^H(j)}$) to its hyperdegree until the size of the hyperedge e_i reaches $d^E(e_i)$. Specifically, duplicated nodes are ignored in each hyperedge generation. The algorithm is terminated until the size of each hyperedge reaches the pre-set size.

In the HyperCL, the hyperdegree sequence is generated by a hyperdegree distribution $p(d^H) \sim (d^H)^{-\Theta}$, where the exponent Θ is a tunable parameter. As the value of exponent Θ increases, the hyperdegree distribution would change from heterogeneous to homogeneous. In this work, the exponent value is set as $\Theta = 2, 2.1, 2.3$ and 2.5. The hyperedge size sequence generated by a uniform distribution with the maximal size setting as 10, respectively. Coefficient of variation (CV), defined as the ratio of the

Table 4
AUC scores obtained by our algorithms and baselines. The best performance, i.e., largest AUC score, is shown by ** and the second best is shown by * in each hypergraph. We set $\beta = 0.015, T = 15$.

Hypergraphs	Algorithms (AUC)							
	HADP	HSDP	H-RIS	H-CI ($l = 1$)	H-CI ($l = 2$)	H-IMRANK	H-Degree	Degree
Algebra	**0.1993****	0.1532*	0.1337	0.1009	0.0971	0.1081	0.0987	0.1090
Restaurant-Rev	0.1364*	0.1276	**0.1385****	0.1185	0.1106	0.1264	0.1149	0.1271
Geometry	**0.1348****	0.1306	0.1312*	0.1200	0.1202	0.1226	0.1200	0.1207
Music-Rev	0.1320*	0.1279	**0.1397****	0.1211	0.1156	0.1184	0.1184	0.1268
NDC-classes	0.1426*	**0.1526****	0.0889	0.1186	0.1198	0.1331	0.1201	0.1244
Bars-Rev	**0.1263****	0.1261*	0.1259	0.1245	0.1236	0.1240	0.1240	0.1256
iAF1260b	**0.2078****	0.1426*	0.1030	0.1023	0.1002	0.1368	0.1006	0.1068
iJO1366	**0.1946****	0.1425	0.1527	0.0882	0.0830	0.1584*	0.0837	0.0967

Table 5
AUC scores obtained by our algorithms and baselines. The best performance, i.e., largest AUC score, is shown by ** and the second best is shown by * in each hypergraph. We set $\beta = 0.02, T = 10$.

Hypergraphs	Algorithms (AUC)							
	HADP	HSDP	H-RIS	H-CI ($l = 1$)	H-CI ($l = 2$)	H-IMRANK	H-Degree	Degree
Algebra	**0.2095****	0.1536*	0.1417	0.0975	0.0925	0.1040	0.0957	0.1053
Restaurant-Rev	0.1383*	0.1271	**0.1449****	0.1168	0.1078	0.1254	0.1129	0.1266
Geometry	**0.1405****	0.1340*	0.1329	0.1174	0.1176	0.1216	0.1173	0.1186
Music-Rev	**0.1409****	0.1314	0.1387*	0.1181	0.1111	0.1155	0.1146	0.1296
NDC-classes	0.1365*	**0.1467****	0.1093	0.1170	0.1183	0.1310	0.1186	0.1227
Bars-Rev	0.1268*	**0.1270****	0.1221	0.1253	0.1232	0.1251	0.1239	0.1267
iAF1260b	**0.2040****	0.1412*	0.1040	0.1035	0.1018	0.1359	0.1021	0.1076
iJO1366	**0.2220****	0.1550	0.0831	0.0913	0.0867	0.1728*	0.0880	0.1011

Table 6
Time cost for each algorithm. The running time are given by the average over 10 realizations, the seed set size is set as $K = 25$. The best performance, i.e., minimal time cost, is shown by * and the running time of our proposed algorithm is shown in bold in each hypergraph. We set $\beta = 0.01, T = 25$.

Hypergraphs	Time cost (s)								
	HADP	HSDP	H-RIS	H-CI ($l = 1$)	H-CI ($l = 2$)	Greedy	H-IMRANK	H-Degree	Degree
Algebra	**19.2191**	1.7816	101.4887	1.4566	489.7527	15 991.0652	17.9884	0.0290	0.0216*
Restaurant-Rev	**9.9571**	1.5490	79.2794	1.1639	219.7846	23 630.6722	71.0967	0.0434	0.0311*
Geometry	**46.6095**	2.6269	173.1826	2.6695	2369.7399	53 966.0565	44.9712	0.0293*	0.0294
Music-Rev	**30.4322**	3.4626	618.9846	3.6286	2164.4441	144 976.2404	316.2436	0.0623*	0.0653
NDC-classes	**8.7360**	2.9378	4317.1805	1.3690	5748.8244	18 891.5252	73.2828	0.0560*	0.0637
Bar-Rev	**30.9617**	3.6873	3472.9475	3.9713	12 715.2541	131 718.2580	574.7956	0.0780	0.0621*
iAF1260b	**14.8016**	4.1104	3532.0354	1.9957	92 402.2618	15 396.0684	229.3660	0.1050	0.0885*
iJO1366	**19.3894**	4.5724	9123.7571	2.2732	91 108.2699	30 233.7775	257.6366	0.0824*	0.0943

standard deviation to the mean (Tanaka, 2005; Zhang et al., 2019), is utilized to measure the degree heterogeneity of a hypergraph. Specifically, both the standard deviation and the mean are obtained from the node degree sequence. Furthermore, we show the correlation between the degree and hyperdegree of a node in the synthetic hypergraphs generated by HyperCL in Fig. 6, where the PCC is higher than 0.9 in hypergraphs generated by different hyperdegree distribution.

We show the performance of our methods and the baselines on synthetic hypergraphs on IM problem in Fig. 7 and Table 7, respectively. We observe that HADP surpasses all the other methods in all the hypergraphs and H-RIS performs the second best. In addition, as Θ decreases, i.e., the degree distribution is more heterogeneous, HADP can gain more improvement in AUC than H-RIS (Table 7). It suggests that HADP tends to be more suitable for solving HIM with heterogeneous degree distribution, which is common in real world. For a hypergraph with high degree heterogeneity, nodes with high degree tend to have more neighbors in the seed node set compared to low degree nodes. It implies that HADP could impose a large penalty on high degree nodes in hypergraphs with high degree heterogeneity. Thus, the adaptive degree distribution after several iterations could be significantly different from the original one. However, in hypergraphs with low degree heterogeneity, the degree of nodes tends to be similar, which leads to small difference in penalties of the nodes' degree. That is to say, the adaptive degree distribution after several iterations could be similar to the original one. We have verified the above description by plotting the adaptive degree distributions of different iterations. Therefore, we suggest that the relatively homogeneous degree distribution may lead to HADP performs worse in hypergraphs with low degree heterogeneity for IM problem.

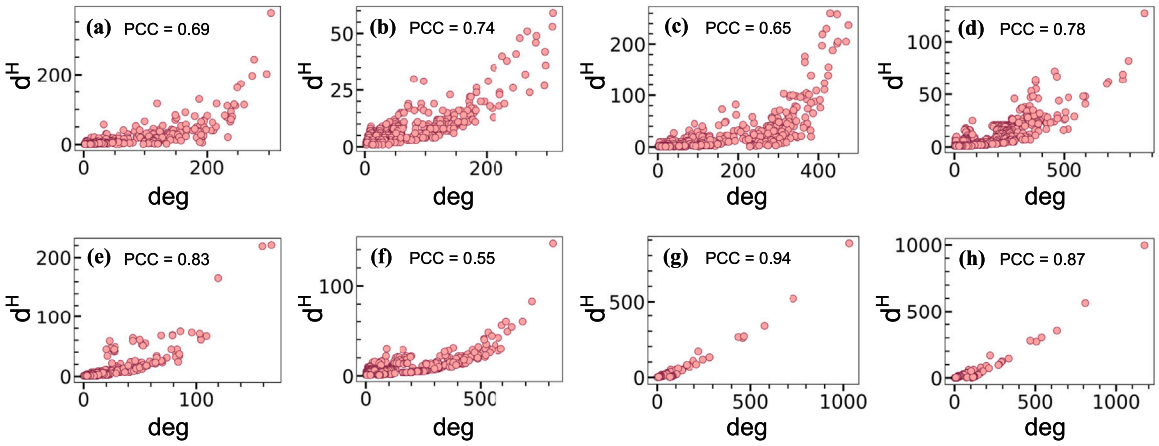


Fig. 5. The correlation between node degree and hyperdegree in hypergraphs generated by real-world datasets: (a) Algebra; (b) Restaurant-Rev; (c) Geometry; (d) Music-Rev; (e) NDC-classes; (f) Bars-Rev; (g) iAF1260b; (h) iJO1366. In each figure, we show the Pearson correlation coefficient (PCC) between node degree and hyperdegree in each of the hypergraphs.

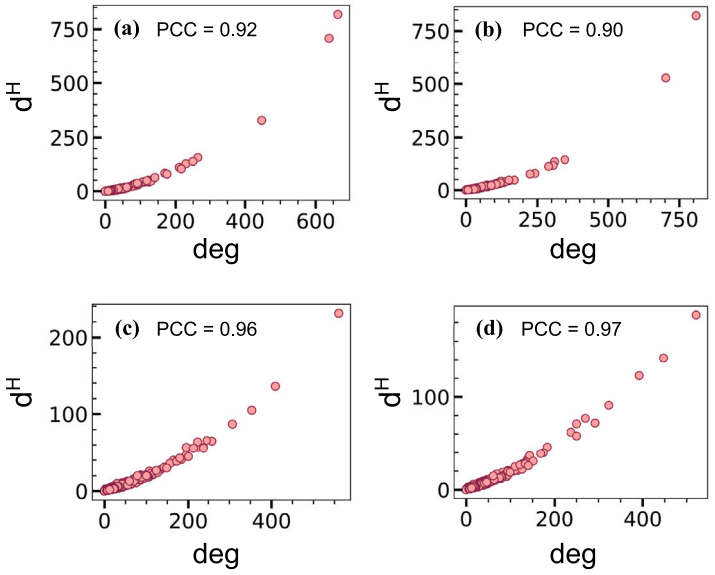


Fig. 6. The correlation between node degree and hyperdegree in synthetic hypergraphs generated by HyperCL via different exponents: (a) $H(\Theta = 2)$; (b) $H(\Theta = 2.1)$; (c) $H(\Theta = 2.3)$; (d) $H(\Theta = 2.5)$. In each figure, we show the Pearson correlation coefficient (PCC) between node degree and hyperdegree in each of the hypergraph, where $n = 1000$ and $m = 1000$.

6. Discussions

6.1. Findings

From the perspective of hypergraph, this study design a spreading mechanism based on the higher-order interactions between entities and solve the HIM under the spreading dynamics. We find that the proposed method, i.e., Hyper Adaptive Degree Pruning (HADP), enables to effectively select seeds that can lead to large influence coverage, especially in hypergraphs with nodes having high influence overlaps with its neighboring nodes. The experiments conducted on various synthetic hypergraphs reveal that HADP performs better in terms of effectiveness and robustness in hypergraphs with high degree heterogeneity (Gao et al., 2022; Li, Ni et al., 2022), which is prevalent in real-world systems.

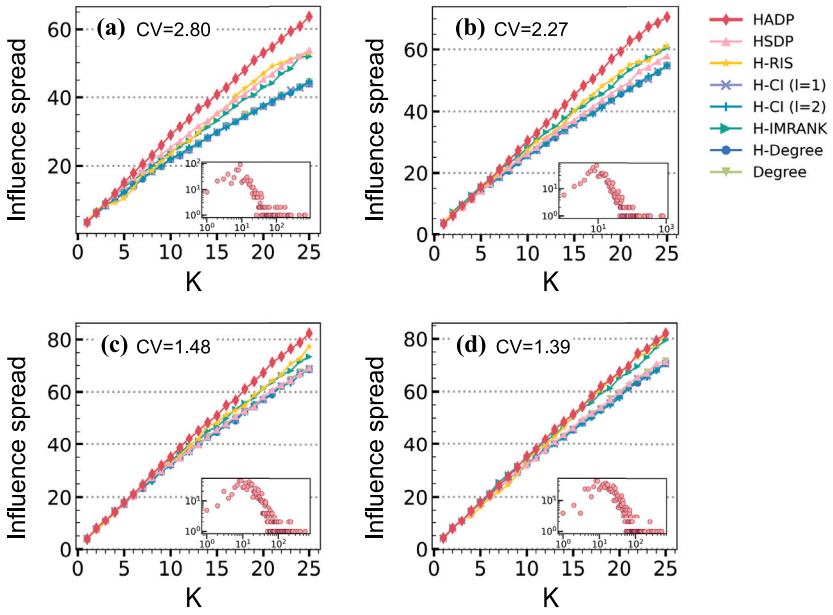


Fig. 7. Expected influence spread as a function of K for each algorithm on synthetic hypergraphs generated by HyperCL: (a) $H(\Theta = 2)$; (b) $H(\Theta = 2.1)$; (c) $H(\Theta = 2.3)$; (d) $H(\Theta = 2.5)$. The inset in each figure depicts the degree distribution of each hypergraph, where $n = 1000$ and $m = 1000$.

6.2. Theoretical contributions

This study sheds light on the problem of multiple influential nodes identification in hypergraphs. Previously, only a few studies have addressed this problem on higher-order networks.

In our work, a seed node screening strategy HADP considering node adaptive degree with higher-order interactions is proposed. The algorithm fully considers the overlap of its affiliated hyperedges when performing degree pruning. When the two-hop neighbors of a node are adjacent to those one-hop ones in more hyperedges, our method will have a greater degree reduction for each one-hop neighbor of the candidate node, which is the major difference from previous studies. Additionally, the proposed algorithm may inspire the studies of network dismantling (Wandelt et al., 2018) and influence minimization (Wang et al., 2017) problems in higher-order networks.

In the design of HADP, we have analyzed some of the properties of a hypergraph in detail, which may help us deepen the understanding of the nature and topology of higher-order networks. In particular, we observe a high influence overlap between nodes and its neighbors in most of the hypergraphs when conducting the spreading process on them. Besides, a strong correlation between node degree and hyperdegree in empirical hypergraphs is observed.

Based on the properties revealed, we measure the effectiveness of our algorithm on synthetic hypergraphs with different degree heterogeneity. It is concluded that our algorithm performs better in networks with higher heterogeneity, which provides theoretical support for the possible applicability of the algorithm to tackle the HIM problem in most of the real systems with high degree heterogeneity.

6.3. Practical significance

Since entities in many real systems contain higher-order interactions like hyperedges rather than simple links, the approach in this paper could be applied to marketing, epidemic prevention and social opinion management, etc. For example, with regard to the operation and management of user comments on social platforms, relevant management departments should focus more on the comments and dynamics of bloggers with high influence in order to quickly inhibit the spread of rumors and guarantee the authenticity of online social opinions.

7. Conclusions

Much effort has been devoted to find influential node set in ordinary networks. In this work, we tackle the challenge on IM problem in hypergraphs, which aims to identify K initial spreaders from a hypergraph that can maximize the expected outbreak size of a certain spreading dynamics. We start with a simple spreading model, i.e., susceptible–infected (SI) model with contact process dynamics. Based on the fact that the influence overlap between nodes and their neighbors is usually high in hypergraphs generated by real data, we propose an algorithm called Hyper Adaptive Degree Pruning (HADP) to solve the HIM problem. The algorithm

Table 7
AUC scores obtained by each of the curve presented in Fig. 7 for algorithms, i.e., HADP, HSDP, H-RIS, H-CI ($l = 1$), H-CI ($l = 2$), H-degree and Degree. The best performance, i.e., the largest AUC score, is shown by $**$ and the second best is shown by $*$ in each hypergraph.

Hypergraphs	Algorithms (AUC)								CV	Gain
	HADP	HSDP	H-RIS	H-CI ($l = 1$)	H-CI ($l = 2$)	H-IMRANK	H-Degree	Degree		
$H(\Theta = 2)$	**0.1540****	0.1343*	0.1317	0.1128	0.1130	0.1280	0.1127	0.1134	2.80	14.67%
$H(\Theta = 2.1)$	**0.1467****	0.1227	0.1302	0.1173	0.1173	0.1306*	0.1173	0.1179	2.27	12.33%
$H(\Theta = 2.3)$	**0.1380****	0.1220	0.1288*	0.1210	0.1207	0.1274	0.1206	0.1216	1.48	7.14%
$H(\Theta = 2.5)$	**0.1349****	0.1220	0.1312	0.1202	0.1195	0.1314*	0.1202	0.1206	1.39	2.66%

iteratively gives large penalty to nodes that have more neighbors in the existing seed set and thus these nodes are less likely to be chosen as seeds. To validate the algorithm effectiveness, we demonstrate a list of baseline algorithms, including the ones proposed by previous researchers as well as algorithms extended from ordinary networks. We perform the experiments on eight hypergraphs generated by real data from various domains. Results show that HADP is superior to the benchmarks in terms of accuracy (except Greedy) almost in all the hypergraphs with different infection probability. In addition, our algorithm also shows good performance in terms of efficiency. As HADP is based on the node degree, we further test the performance on synthetic hypergraphs generated by HyperCL, which can generate hypergraphs with different hyperdegrees. The results demonstrate HADP gains more AUC scores in hypergraphs with high degree heterogeneity.

Heuristic algorithms have been widely utilized to solve the IM on ordinary networks due to its low computational complexity. In this work, we confine to use a simple heuristic from hypergraph, i.e., degree, to design algorithm for identifying seed node set, which shows high performance. We deem that more high-order properties from hypergraph could be used for IM. Moreover, our algorithm framework could also be promising in solve the IM problem for other dynamic processes, such as threshold model (Xu et al., 2022), independent cascade model (Ma & Rajkumar, 2022) and other epidemic models (Jhun et al., 2019).

CRediT authorship contribution statement

Xiu-Xiu Zhan: Planed the study, Performed the experiments and prepared the figures, Analyzed the results and wrote the manuscript, Read and approved the final. **Chuang Liu:** Planed the study, Performed the experiments and prepared the figures, Analyzed the results and wrote the manuscript, Read and approved the final. **Zi-Ke Zhang:** Planed the study, Performed the experiments and prepared the figures, Analyzed the results and wrote the manuscript, Read and approved the final.

Data and code availability

The data and codes used for this study are available at https://github.com/DDMXIE/Influence-maximization-on-hypergraphs.

Acknowledgments

This work was supported by Natural Science Foundation of Zhejiang Province, China (Grant Nos. LQ22F030008 and LR18A050001), the National Natural Science Foundation of China (Grant Nos. 92146001, 61873080 and 61673151), the Major Project of The National Social Science Fund of China (Grant No. 19ZDA324), the Scientific Research Foundation for Scholars of HZNU (2021QDL030) and the Fundamental Research Funds for the Central Universities.

References

Amato, F., Moscato, V., Picariello, A., & Sperlí, G. (2017). Influence maximization in social media networks using hypergraphs. In *International conference on green, pervasive, and cloud computing* (pp. 207–221). Springer.

Antelmi, A., Cordasco, G., Spagnuolo, C., & Szufel, P. (2021). Social influence maximization in hypergraphs. *Entropy*, *23*(7), 796.

Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., Young, J.-G., & Petri, G. (2020). Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, *874*, 1–92.

Biswas, T. K., Abbasi, A., & Chakrabortty, R. K. (2021). An MCDM integrated adaptive simulated annealing approach for influence maximization in social networks. *Information Sciences*, *556*, 27–48.

Borgs, C., Brautbar, M., Chayes, J., & Lucier, B. (2014). Maximizing social influence in nearly optimal time. In *Proceedings of the 25th annual ACM-SIAM symposium on discrete algorithms* (pp. 946–957). SIAM.

Bozorgi, A., Haghighi, H., Zahedi, M. S., & Rezvani, M. (2016). INCIM: A community-based algorithm for influence maximization problem under the linear threshold model. *Information Processing & Management*, *52*(6), 1188–1199.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, *30*(1–7), 107–117.

Cencetti, G., Battiston, F., Lepri, B., & Karsai, M. (2021). Temporal properties of higher-order interactions in social networks. *Scientific Reports*, *11*(1), 7028.

Chen, W., Wang, C., & Wang, Y. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1029–1038).

Cheng, C.-H., Kuo, Y.-H., & Zhou, Z. (2020). Outbreak minimization vs influence maximization: an optimization framework. *BMC Medical Informatics and Decision Making*, *20*(1), 266.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 57–66).

Ferraz de Arruda, G., Tizzani, M., & Moreno, Y. (2021). Phase transitions and stability of dynamical processes on hypergraphs. *Communications Physics*, *4*(1), 24.

Gao, L., Wang, H., Zhang, Z., Zhuang, H., & Zhou, B. (2022). HetInf: Social influence prediction with heterogeneous graph neural network. *Frontiers in Physics*, 729.

Gong, Y., Liu, S., & Bai, Y. (2021). Efficient parallel computing on the game theory-aware robust influence maximization problem. *Knowledge-Based Systems*, *220*, Article 106942.

Goyal, A., Lu, W., & Lakshmanan, L. V. (2011). Celf++ optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on world wide web* (pp. 47–48).

Huang, H., Shen, H., Meng, Z., Chang, H., & He, H. (2019). Community-based influence maximization for viral marketing. *Applied Intelligence*, *49*(6), 2137–2150.

Jhun, B., Jo, M., & Kahng, B. (2019). Simplicial SIS model in scale-free uniform hypergraph. *Journal of Statistical Mechanics: Theory and Experiment*, *2019*(12), Article 123207.

Jung, K., Heo, W., & Chen, W. (2012). Irie: Scalable and robust influence maximization in social networks. In *2012 IEEE 12th international conference on data mining* (pp. 918–923). IEEE.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 137–146).

Kumar, S., Singhla, L., Jindal, K., Grover, K., & Panda, B. (2021). IM-ELPR: Influence maximization in social networks using label propagation based community structure. *Applied Intelligence*, *51*(11), 7647–7665.

Lee, G., Choe, M., & Shin, K. (2021). How do hyperedges overlap in real-world hypergraphs?-patterns, measures, and generators. In *Proceedings of the web conference 2021* (pp. 3396–3407).

Lei, S., Maniu, S., Mo, L., Cheng, R., & Senellart, P. (2015). Online influence maximization. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 645–654).

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 420–429).

Li, W., Li, Y., Liu, W., & Wang, C. (2022). An influence maximization method based on crowd emotion under an emotion-based attribute social network. *Information Processing & Management*, *59*(2), Article 102818.

Li, W., Ni, L., Wang, J., & Wang, C. (2022). Collaborative representation learning for nodes and relations via heterogeneous graph neural network. *Knowledge-Based Systems*, *255*, Article 109673.

Li, W., Zhong, K., Wang, J., & Chen, D. (2021). A dynamic algorithm based on cohesive entropy for influence maximization in social networks. *Expert Systems with Applications*, *169*, Article 114207.

Lü, L., Chen, D., Ren, X.-L., Zhang, Q.-M., Zhang, Y.-C., & Zhou, T. (2016). Vital nodes identification in complex networks. *Physics Reports*, *650*, 1–63.

Ma, A., & Rajkumar, A. (2022). Hyper-IMRANK: Ranking-based influence maximization for hypergraphs. In *5th joint international conference on data science & management of data (9th ACM IKDD CODS and 27th COMAD)* (pp. 100–104).

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

Ouvrard, X. (2020). Hypergraphs: an introduction and review. arXiv preprint arXiv:2002.05014.

Qiu, L., Tian, X., Zhang, J., Gu, C., & Sai, S. (2021). LIDDE: A differential evolution algorithm based on local-influence-descending search strategy for influence maximization in social networks. *Journal of Network and Computer Applications*, *178*, Article 102973.

Samir, A. M., Rady, S., & Gharib, T. F. (2021). LKG: A fast scalable community-based approach for influence maximization problem in social networks. *Physica A: Statistical Mechanics and its Applications*, *582*, Article 126258.

Singh, S. S., Kumar, A., Singh, K., & Biswas, B. (2019). C2IM: Community based context-aware influence maximization in social networks. *Physica A: Statistical Mechanics and its Applications*, *514*, 796–818.

Singh, S. S., Srivastva, D., Verma, M., & Singh, J. (2021). Influence maximization frameworks, performance, challenges and directions on social network: A theoretical study. *Journal of King Saud University-Computer and Information Sciences*, (1319–1578).

St-Onge, G., Iacopini, I., Latora, V., Barrat, A., Petri, G., Allard, A., & Hébert-Dufresne, L. (2022). Influential groups for seeding and sustaining nonlinear contagion in heterogeneous hypergraphs. *Communications Physics*, *5*(1), 25.

Stegehuis, C., & Peron, T. (2021). Network processes on clique-networks with high average degree: the limited effect of higher-order structure. *Journal of Physics: Complexity*, *2*(4), Article 045011.

Suo, Q., Guo, J.-L., & Shen, A.-Z. (2018). Information spreading dynamics in hypernetworks. *Physica A: Statistical Mechanics and its Applications*, *495*, 475–487.

Tanaka, R. (2005). Scale-rich metabolic networks. *Physical Review Letters*, *94*(16), Article 168101.

Wandelt, S., Sun, X., Feng, D., Zanin, M., & Havlin, S. (2018). A comparative analysis of approaches to network-dismantling. *Scientific Reports*, *8*(1), 13513.

Wang, B., Chen, G., Fu, L., Song, L., & Wang, X. (2017). Drimux: Dynamic rumor influence minimization with user experience in social networks. *IEEE Transactions on Knowledge and Data Engineering*, *29*(10), 2168–2181.

Wang, C., Chen, W., & Wang, Y. (2012). Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, *25*(3), 545–576.

Wang, J., Ma, X.-J., Xiang, B.-B., Bao, Z.-K., & Zhang, H.-F. (2022). Maximizing influence in social networks by distinguishing the roles of seeds. *Physica A: Statistical Mechanics and its Applications*, Article 127881.

Wang, Z., Sun, C., Xi, J., & Li, X. (2021). Influence maximization in social graphs based on community structure and node coverage gain. *Future Generation Computer Systems*, *118*, 327–338.

Xu, X.-J., He, S., & Zhang, L.-J. (2022). Dynamics of the threshold model on hypergraphs. *Chaos. An Interdisciplinary Journal of Nonlinear Science*, *32*(2), Article 023125.

Young, J.-G., Petri, G., & Peixoto, T. P. (2021). Hypergraph reconstruction from network data. *Communications Physics*, *4*(1), 135.

Zhan, X.-X., Li, Z., Masuda, N., Holme, P., & Wang, H. (2020). Susceptible-Infected-Spreading-based network embedding in static and temporal networks. *EPJ Data Science*, *9*(1), 30.

Zhang, Y., Shi, Z., Feng, D., & Zhan, X.-X. (2019). Degree-biased random walk for large-scale network embedding. *Future Generation Computer Systems*, *100*, 198–209.

Zhu, J., Zhu, J., Ghosh, S., Wu, W., & Yuan, J. (2018). Social influence maximization in hypergraph in social networks. *IEEE Transactions on Network Science and Engineering*, *6*(4), 801–811.