Physica A 526 (2019) 120902

Contents lists available at ScienceDirect

Physica A

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

MIM2: Multiple influence maximization across multiple social networks

Shashank Sheshar Singh *, Kuldeep Singh, Ajay Kumar, Bhaskar Biswas

Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, 221–005, India

HIGHLIGHTS

- A novel MIM2 algorithm is proposed to maximize influence spread across multiple social networks.
- The algorithm is self-decisive in finding the budget of each product.
- The algorithm relies on multiple products and multiple networks simultaneously.
- Classical Linear Threshold and Independent Cascade diffusion models are utilized.
- The proposed algorithm is a trade-off between quality and efficiency.

ARTICLE INFO

Article history:
Received 17 December 2018
Received in revised form 10 February 2019
Available online 10 April 2019

Keywords:
Influence maximization
Multiple influence maximization
Information diffusion
Multiplex social networks

ABSTRACT

Influence maximization (IM) is the problem of selecting a small subset of users with the aim of maximizing influence spread to help marketers in promoting their products. None of the existing literature considers the scenario that a marketing company wants to promote multiple products in multiple networks or a network with the different channel of interactions simultaneously. Considering this scenario, we introduce *multiple influence maximization across multiple social networks* (MIM2) problem. This problem considers the assumption that an influential user can accept multiple products for free and non-influential users have enough purchasing power to adopt multiple promotions from their social interactions. It is also important to consider the role of overlapping users to spread the influence across networks. To address these issues, we propose a unified framework to analyze and represent the MIM2 problem. More specifically, first, we perform a mapping to couple a set of networks into a multiplex network via direct linkage strategy. Second, we propose a heuristic method to find the most influential user over multiple product diffusion multiplex networks. Third, we prove that MIM2 problem is NP-hard and expected influence spread function is sub-modular under traditional diffusion models. Finally, the experimental results show that the advantage of proposed IM problem over existing IM problems.

1. Introduction

In the recent decade, the online social networks like Facebook, Flixster, Twitter, Google+, and Myspace, etc., provide a platform for user's interaction and communication, marketing and promotion for the new product, idea, and innovation. These social networks diffuse and spread information of a product, news, and innovation between users via links using

* Corresponding author.
E-mail addresses: shashankss.rs.cse16@iitbhu.ac.in (S.S. Singh), kuldeep.rs.cse13@iitbhu.ac.in (K. Singh), ajayk.rs.cse16@iitbhu.ac.in (A. Kumar), bhaskar.cse@iitbhu.ac.in (B. Biswas).

https://doi.org/10.1016/j.physa.2019.04.138

the *word-of-mouth* effect [1]. The word-of-mouth advertising leads to an immense application potential in viral marketing [2], rumor control [3,4], revenue maximization [5], social recommendation [6], and network monitoring [7], etc. Inspired by the idea of viral marketing, Pedro and Matt [2] were first to introduced *Influence Maximization* (IM) as an optimization problem. IM problem finds a set of k most influential users in a social network so that the aggregate influence is maximized. Kempe et al. [8] proved that IM problem is NP-hard under traditional diffusion models. To solve IM problem, they introduced a greedy algorithm and proved that the greedy solution is approximated to within a factor of $(1 - 1/e - \varepsilon)$.

Nearly most of the previous research works only focused on network topology and ignored some critical factors like multiple product advertisement, users engagement across networks,[1] different channels of interaction, and network of networks etc. For example, we consider a scenario that an advertising company wants to promote multiple products across multiple social networks simultaneously. In real-world, distinct users have different influence or interest for distinct product in social networks. Thus, each user have different influence probabilities from their neighbors. Nowadays, many users in networks are actively involved in multiple networks simultaneously. Therefore, *overlapping users*[2] play a pivot role in information spreading across networks. Thus, the study of IM problem in each network separately underestimates the social influence of overlapping users in other networks. Hence, an advertising company needs a promotional strategy that seed users can advocate multiple products together and non-seed users can accept different products across networks simultaneously.

Correspondingly, we propose a framework *multiple influence maximization across multiple social networks* (MIM2) for the above scenario. Compared to traditional IM approaches, multiple influence maximization across multiple social networks introduces several new challenges:

1. How to fix the number of items for each product from budget k and how to finds seed users S, $|S| \leq k$.
2. How to identify overlapping users on different social networks and how to coupled these networks into a single one. Also, how to measure the social influence of overlapping and non-overlapping users accurately in a coupled network for each product.
3. How to decide, for which product a user is easily influenced and in which network. Also, in which network information is propagated better.

To tackle these challenges, we present a novel framework to illustrate multiple influence maximization across multiple social networks.

Contribution. To the best of our knowledge, we are the first to introduce MIM2 problem. The major contributions of our work are as follow.

- We propose a novel framework MIM2 for influence maximization across multiple social networks over multiple products simultaneously.
- We prove that MIM2 problem is NP-hard and expected influence spread $\sigma(S)$ is sub-modular under traditional diffusion models.
- We present a model representation for MIM2 problem equivalent to classical IM problem. We also present various coupling strategy to form a multiplex network from multiple single network. The proposed coupling methods can be applied to traditional diffusion models.
- We propose a novel MIM2 algorithm to find seed set across multiple networks with multiple products simultaneously. It also fixes the budget for each product to maximize influence spread.
- We perform experimental analysis to validate the efficiency and influence spread of the proposed algorithm. It also covers the advantage of MIM2 framework over classical IM problem.

Paper organization. The rest of the paper is organized as follows. Section 2 presents a discussion on related work of the IM problem. Section 3 describes primary information regarding problem formulation. Section 4 presents graph notations, influence propagation model, and problem definition. Section 5 explains the proposed MIM2 framework. Section 6 discusses the algorithm with an analysis of their time complexity. Section 7 explains the experimental setup and result analysis of real-word social networks. Section 8 draws the conclusion and future directions.

2. Literature review

Most of the existing research works studied the different variant of IM problem for a single product on separate social networks. Kempe et al. [8] firstly formatted the IM problem and introduced a greedy framework to solve the IM problem. They proposed two widely known influence propagation model, *independent cascade* (IC) model and *linear threshold* (LT) model, where information disseminates in discrete steps, further treated it as a discrete optimization problem and proved it is NP-hard for both two propagation model. Simultaneously, they worked out the greedy algorithm framework with an approximate ratio of $(1 - 1/e)$. Svirdenko et al. [9] extends IM problem defined by [8]

[1] Networks and social networks are interchangeably used in this article.

[2] Users who actively engage in multiple networks simultaneously.

Table 1
Notations.

$G(V, E, W)$	$\triangleq$	A social network with vertex set V and edge set E with edge weight set W
$N(u)$	$\triangleq$	The neighbors set of node u
V_a	$\triangleq$	The set of active nodes
S	$\triangleq$	Seed set
k	$\triangleq$	The number of nodes in seed set ($\|S\|$)
$p_{x,y}$	$\triangleq$	The influence probability of x to adjacent node y.
$\sigma(S)$	$\triangleq$	The expected influence of seed set in the network, i.e., $Inf(S)$
m	$\triangleq$	The number of products for advertising
l	$\triangleq$	The number of channel of interaction or relationship
L	$\triangleq$	The set of relationships

Table 2
Abbreviations.

IM	$\triangleq$	Influence Maximization
IM2	$\triangleq$	Influence Maximization across Multiple social networks
MIM	$\triangleq$	Multiple Influence Maximization
MIM2	$\triangleq$	Multiple Influence Maximization across Multiple social networks
IC	$\triangleq$	Independent Cascade model
LT	$\triangleq$	Linear Threshold model

considering non-uniform selection cost. Based on node potential and sub-modular property of the objective function, some more variants of greedy algorithm are introduced [10–12]. The greedy algorithm has low efficiency due to time-consuming Monte-Carlo (MC) simulations and can hardly adapt to large scale online social network, there are so many works to accelerate influence computation and seed selection, such as heuristic-based approaches [10,13,14], influence-path based approaches [15–17], community-based approaches [18–20], sampling based [21–23], diffusion probabilities estimation [24,25], and score estimation based [26–28].

Some of the users in online social networks participate in multiple social networks simultaneously. Therefore, IM approaches of a single network ignore the social influence of users in other participating networks. Hence, these approaches are not able to estimate the social influence of users accurately. Inspired by the above weakness of IM approaches, researchers have started to explore *influence maximization on multiple networks* (IM2). In order to propagate information across social networks, there is a need for a network coupling strategy. Liu et al. [29] presented a coupling strategy to form multiplex. The authors focused on understanding the flow of information and network clustering. Yagan et al. [30] investigated the information diffusion process in over-laying networks. They investigated the outbreak of information using the SIR model on random networks. Considering users involvement and interest, Shen et al. [31] presented information propagation strategy for multiplex networks. The authors use a lossy coupling scheme and ignore individual network properties. Zhang et al. [32] proposed an improved greedy algorithm for influence maximization across multiple social networks. They also proposed new lossless and lossy coupling schemes for linking multiple social networks effectively.

Inspired by the idea that IM for multiple products simultaneously, Sun et al. [33] proposed *multiple influence maximization* (MIM) problem in a single network. They consider a separate product diffusion graph for each product. The authors adopt a greedy framework for seed selection. They assume that the influence of a node is limited to six-hope area. The experimental results show that the advantage of MIM over classical IM problem. Recently, Context-aware influence maximization techniques are introduced to improve the effectiveness of seed. Some research works have been proposed considering contextual features such as location-aware [34,35], time-aware [36,37], topic-aware [38–40], and competitive [41,42]. Inspired by IM2 and MIM, we present a novel framework *multiple influence maximization across multiple social networks* (MIM2). To the best of our knowledge, this is the first work considering MIM2 framework.

3. Preliminaries

3.1. Notations

Notations and Abbreviations that are needed for problem formulations in this paper are given in Tables 1 and 2 respectively.

3.2. Definitions

Definition 1 (*Social Network*)**.** A social network with N users and M social ties is represented as a weighted-directed graph $G(V, E, W)$. Here, V denotes set of users, $|V| = N$, and E represents a set of relationships, $|E| = M$, and W represents edge-weight. A social network is also known as an influence graph.

Definition 2 (*Neighbors*)**.** Neighbors $N(u)$ of node u is defined as the set of users v such that $v \in N(u)$ iff $\exists(u, v) \in E$, $v \in V$. $N_{inc}(u)$ and $N_{out}(u)$ denote in and out neighbors of node u respectively.

Definition 3 (*Degree Centrality*)**.** Degree centrality is defined as the number of links incident upon a node i.e. $C_D(u) = |N(u)|$. In directed social network, degree of u is considered as $C_D(u) = |N_{out}(u)|$.

Definition 4 (*Seed Nodes*)**.** Seed nodes (S) are the set of nodes who act as the source of the information propagation process in the social network, $|S| = k$, $S \in V$.

Definition 5 (*Active Node*)**.** A node $u \in V$ is called active if either $u \in S$ or u adopted the information propagated by previously active nodes $v \in V_A$ under diffusion model. Once u is activated, then $V_a \leftarrow \{V_A \cup u\}$.

Definition 6 (*Influence Spread*)**.** Influence spread $I_S(S)$ of the seed set S is defined as the number of active users after diffusion process under a diffusion model, i.e., $I_S(S) = |V_A(S)|$.

Definition 7 (*Information Diffusion Model (IDM/DM) [43]*)**.** Given an influence graph $G = (V, E, W)$, a seed set $S \subseteq V$ and an information diffusion model captures the stochastic process for S influence spreading on graph G.

Definition 8 (*Influence Maximization (IM) [8]*)**.** Given an influence graph $G = (V, E, W)$, an information diffusion model, a positive integer k, then influence maximization process selects a seed set $S \subseteq V$ of k users to maximize the influence spread in G, i.e., $\sigma(S) = argmax_{S* \subseteq V \wedge |S*|=k} \sigma(S^*)$.

3.3. Diffusion model

Kempe et al. [8] incorporated two basic diffusion models, *linear threshold* (LT) and *independent cascade* (IC) for information propagation. In both of these models, each node at any time-stamp t belongs to one of the two state: inactive and active. Nodes that are not influenced by their neighbors or not heard about the product are known as inactive nodes. Initially at time $(t = 0)$, all nodes are inactive. Active nodes are influenced by their neighbors and only such nodes can propagate influence to their neighbors.

3.3.1. Linear threshold model (LT)

In this model, every node x has an activation threshold θ_u and a node y becomes active only if $\Sigma_{x \in N^A_{inc}(y)} w(x, y) \geq \theta_x$ where $N^A_{inc}(y)$ and $w(x, y)$ are set of active incoming neighbors of y and edge weight of (x, y) respectively. For each edge $(x, y) \in E$, we assign edge weight $w(x, y)$ in LT model as follows.

$$w(x, y) = \begin{cases} \dfrac{1}{indegree(y)} & \text{Uniform} \\ (0, 1) & \text{Random} \\ \dfrac{c(x, y)}{\Sigma c(x, y)} & \text{Parallel} \end{cases} \tag{1}$$

where $c(x, y)$ is the number of parallel edges from x to y in multi-graph.

3.3.2. Independent cascade model (IC)

In this model, when a node x becomes active at time t, it has a only chance to activate its inactive neighbors y with activation probability p_{xy} at the time $(t + 1)$. If node y becomes active at the time $(t + 1)$ then it will never be inactive in future. The diffusion process terminates if no node is activated at time $(t + 1)$. For each edge $(x, y) \in E$, we assign edge weight $w(x, y)$ in IC model as follows.

$$w(x, y) = \begin{cases} [0.01, 0.1] & \text{Constant} \\ \dfrac{1}{indegree(y)} & \text{Weighted cascade} \\ \{0.001, 0.01, 0.1\} & \text{Tri-valency model} \end{cases} \tag{2}$$

4. Model and problem definition

4.1. Graph notations

Suppose, there are l graphs $G_1, G_2, \ldots, G_l$, each of the graph is represented as $G_i(V_i, E_i, W_i)$, $1 \leq i \leq l$. The set V_i, E_i, and W_i denote the vertex set, edge set, and strength of their relationship respectively in graph G_i. The $N^i_{in}(x)$ and $N^i_{out}(x)$ represent the set of incoming and outgoing neighbors of a user x in network G_i respectively. We modeled these l networks into a single multiplex network $G(V, E, W)$ by a coupling strategy through *overlapping users* to formulate MIM2 problem. The *overlapping users* can be identified using methods in [44–46]. Then, we construct a graph $G^i(V, E, W^i)$, $1 \leq i \leq m$ from G for each product $i \in m$ to propagate each product influence independently.

4.2. Influence propagation model

To compute the influence spread of seed set S in multiplex network G, we incorporated classical diffusion models present in Section 3.3. First, we start the computation of influence spread for IC model. Similar to [8], there are several paths $path_{x,y}$ exist between a pair of nodes x and y. Let $P(path^j_{x,y})$ represents the influence of an individual x to y along a path $path^j_{x,y} = (x = y_1, y_2, \ldots, y_t = y)$, $path^j_{x,y} \in path_{x,y}$, given as follows.

$$P(path^j_{x,y}) = \prod_{i=1}^{i=t-1} p(y_i, y_{i+1}) \tag{3}$$

where $p_{x,y}$ is the influence probability of x to adjacent node y. The influence probability $P(x, y)$ of non-adjacent nodes x to y based on assumption that influence spreads independently along different paths, is given as follows.

$$P(x, y) = 1 - \prod_{path^j_{x,y} \in path_{x,y}} (1 - P(path^j_{x,y})) \tag{4}$$

Now, we can estimate the influence spread of seed set S on the network under IC. The influence spread of node x can be computed by sum up the influence of each reachable path from node x, given as follows.

$$\sigma(x) = 1 + \sum_{y \in V \setminus x} P(x, y) \tag{5}$$

Similarly, we can compute $\sigma(S)$, as follows.

$$\sigma(S) = |S| + \sum_{y \in V \setminus S} P(S, y) \tag{6}$$

4.3. Problem definition

Given l influence graphs $G_i = (V_i, E_i, W_i)$, an information diffusion model, number of products m, and budget k. Influence maximization process selects a seed set $S \in V$ to maximize the influence spread in G, i.e.,

$$\sigma(S) = \underset{S^* \subseteq V \wedge |S^*| = k}{argmax} \ \sigma(S^*) \tag{7}$$

where, $V = \bigcup_{i=1}^{i=l} V_i$, $S^* = \bigcup_{i=1}^{i=m} S^*_i$, $\sum_{i=1}^{i=m} |S^*_i| = k$ and $|S| \leq k$. The objective function $\sigma(S)$ is sub-modular. An arbitrary function is called sub-modular if it satisfies following two properties.

Property 1 (*Monotone Increasing*)**.** *An objective function $\sigma(S)$ is monotone increasing iff $\sigma(S) \leq \sigma(T)$, $S \subset T$.*

Property 2 (*Diminishing Returns*)**.** *An objective function $\sigma(S)$ is diminishing return iff $\sigma(S \cup u) - \sigma(S) \geq \sigma(T \cup u) - \sigma(T)$, $\forall u \in T$ and $S \subset T$.*

Theorem 1. *The expected influence spread function $\sigma(S)$ of MIM2 is sub-modular under traditional diffusion models.*

Proof. In MIM2 problem, we incorporated traditional diffusion models LT and IC from [8] for influence diffusion. We use the same propagation strategy as in LT and IC. MIM2 considers multiple products and multiple networks simultaneously to improve the effectiveness of seed nodes. From problem definition, it is clear that if MIM2 consider $l = 1$ and $m = 1$, then MIM2 problem reduces to original IM problem formulated by [8]. Therefore, diminishing return and monotone increasing properties directly inherited from LT and IC diffusion models. Hence, we can conclude that the expected influence spread $\sigma(S)$ of MIM2 remains sub-modular under traditional diffusion models. □

Theorem 2. *The MIM2 problem under traditional influence diffusion models is NP-hard.*

Proof. As we discussed earlier, the MIM2 problem can be reduced in conventional IM problem presented in [8] by considering the number of networks $l = 1$ and the number of products $m = 1$. As we know IM problem is NP-hard under traditional diffusion models proved by [8]. Hence, we can say that MIM2 is NP-hard under diffusion models. □

5. MIM2 framework

Suppose, an advertising company wants to select a small set of influential users who interact among a finite set of users through different channels of interaction. For example, the Twitter network consists of three types of interaction networks, such as reply, re-tweet, and mention networks. Each interaction network has the same set of users with the

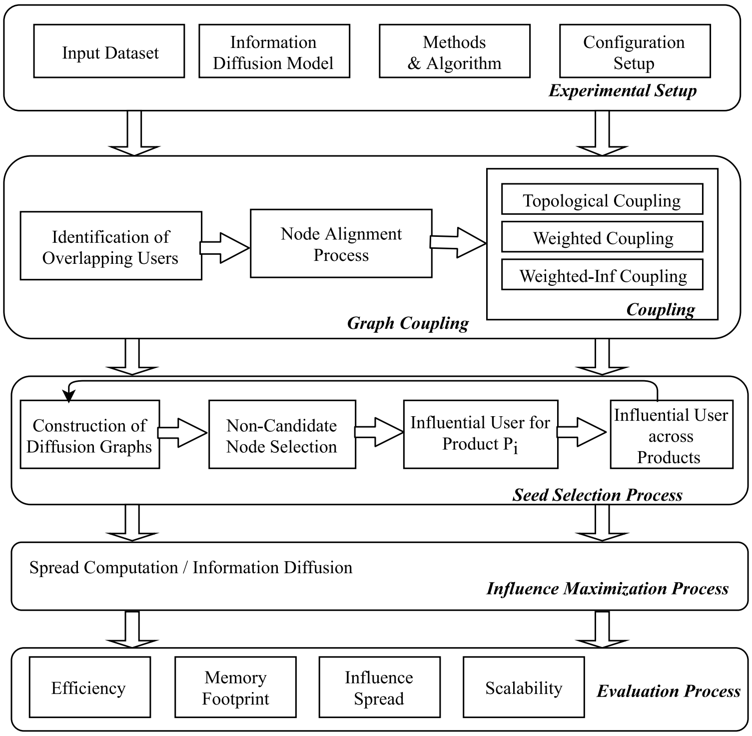


Fig. 1. The MIM2 framework.

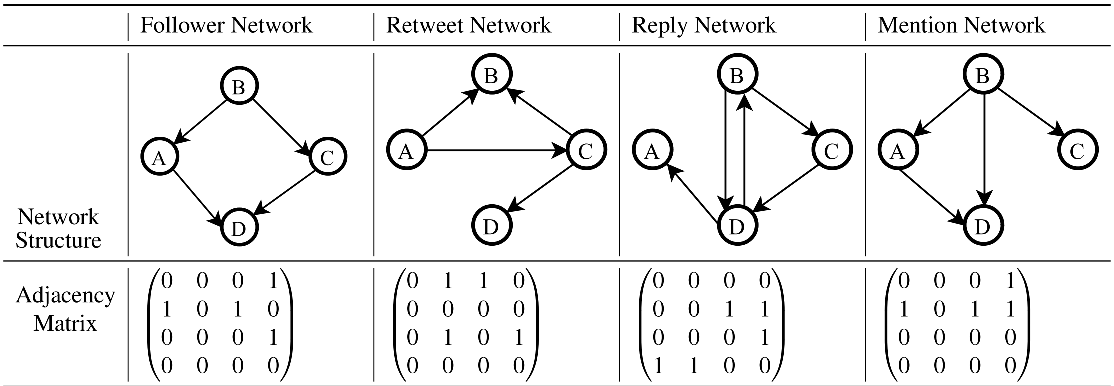


Fig. 2. An example graph of Twitter network (Follower (FN) network) with three type of interaction networks (Retweet (RTN), Reply (REN), and Mention (MEN) networks) with their adjacency matrix.

different channel of interaction as shown in Fig. 2. Fig. 1 presents the proposed MIM2 framework. Let graph $G = (V, E, L)$ represents a complex social network with the different type of relationships, where V is a finite set of users, E is a finite set of relationship between a pair of users with L channels of interaction. The main **Algorithm 1** explains the propose MIM2 framework.

5.1. Identification of overlapping users

MIM2 problem considers users with accounts on multiple social networking sites like Facebook, Twitter, etc., and it brings the opportunity to account the user's influence across multiple networks. This leads to the generation of more effective seed nodes. To perform information diffusion across multiple networks simultaneously, we need to find users who actively involved in multiple networks. These users are useful in graph coupling and known as overlapping or identical users. The identification of overlapping users is a challenging task because of the diversification of users information across networks and also network data is noisy, big, unstructured and incomplete. The overlapping users

Algorithm 1: MIM2 (G_i, k, l, m)

Input: $G_i(V_i, E_i, W_i)$, $1 \leq i \leq l$: Social graphs, k: Size of the seed set, l: Number of social graphs, m: Number of products
Output: S: Seed set

```
S ← ϕ
Remaining ← k
G ← Apply one of the coupling strategy based on Algorithm 2, 3, or 4
Find product diffusion graph G^i of multiplex graph G for each product i, 1 ≤ i ≤ m
while Remaining > 0 do
    seed_node ← NextSeedB(G^i, update_i, Acc_i, seed_i)          ▷ See Algorithm 7
    seed_node.Graph.update ← 0
    seed_node.deleted = True
    S[seed_node.Graph].insert(seed_node)
    Remaining ← Remaining − 1
Return S
```

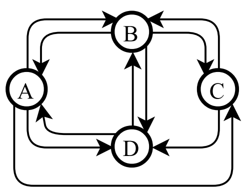
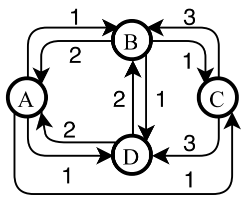
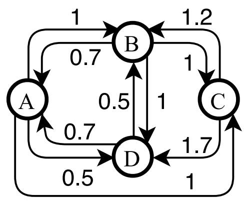
	Topological Coupling	Weighted Coupling	Weighted-influence Coupling
Coupled Multiplex Network			
Multiplex Adjacency Matricx	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 0 & 3 & 0 & 3 \\ 2 & 2 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 & 0.5 \\ 0.7 & 0 & 1 & 0.5 \\ 0 & 1.2 & 0 & 1.7 \\ 0.7 & 1 & 0 & 0 \end{pmatrix}$
Normalized Adjacency Matricx	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.3 & 0.3 & 0.3 \\ 2 & 0 & 0.3 & 0.3 \\ 0 & 1 & 0 & 1 \\ 0.6 & 0.6 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0.5 & 0.5 & 0.2 \\ 0.4 & 0 & 0.5 & 0.2 \\ 0 & 0.7 & 0 & 1 \\ 0.4 & 0.5 & 0 & 0 \end{pmatrix}$

Fig. 3. The coupling of networks.

can be identified using methods present in [44–46]. The identification of overlapping users is not in the focus of this paper. We consider an assumption that a user has a single account in a network.

5.2. Node alignment process

Node alignment is the process of reassigning a *universal identification number* (Uid) to every node u in each network so that every overlapping user has the same Uid across networks. In general, every network has its own node naming system. Therefore, it might be possible that a user has different identification in the different networks. As a consequence, we need to take care of user states across networks repeatedly and it results in extra effort and a complicated mechanism. Therefore, We ease this problem by reassigning a universal Uid to each individual in each network.

5.3. Graph coupling scheme: Direct linkage

In the linkage strategy, each relationship graph $L_i \in L$ is considered as an individual network. In order to transform these relationship networks into a single multiplex network $G = (V, E, W)$, we need to aggregate all relationships between each pair u and v to form a single relationship. Let us consider a complex system $G = (V, E, L)$, $L = \{L_1, L_2, \ldots, L_l\}$ with l types of relationship. Each interaction network $L_i \in L$ is integrated with an adjacency matrix

$A^{L_i} = \{a^{L_i}_{uv}\}$. Therefore, such a complex system can be expressed as $A = [A^{L_1}, A^{L_2}, \ldots, A^{L_l}]$. Similarly, the degree of an individual u can be represented as a vector $D(u) = [D^{L_1}(u), D^{L_2}(u), \ldots, D^{L_l}(u)]$, where $D^{L_i}(u) = \sum_{v \in V} a^{L_i}_{uv}$. Both vector A and $D(u)$ are requisite to represent the complex system correctly. Fig. 2 shows an example Twitter graph with four types of relationship and also presents their corresponding adjacency matrices. From Fig. 2, we can see that $l = 4$ and $D(A) = [1, 2, 0, 1]$.

Algorithm 2: Topological_Coupling($G_i(V_i, E_i, W_i), l$)

Input: $G_i(V_i, E_i, W_i)$, $1 \leq i \leq l$: Social graphs
Output: $G(V, E, W)$: Topological multiplex social graph

$V\_set \leftarrow \phi$
$Edge\_set \leftarrow \phi$
$Weight \leftarrow \phi$
for *each graph* G_i, $1 \leq i \leq l$ **do**
  for *each edge* $(u, v) \in E_i$ **do**
    $Edge\_set \leftarrow Edge\_set \cup (u, v)$
    $V\_set \leftarrow V\_set \cup \{u, v\}$
for *each edge* $(u, v) \in Edge\_set$ **do**
  $Weight[u, v] \leftarrow 1$
Return $G(V\_set, Edge\_set, Weight)$

In order to aggregate these relationships to a single relationship, the direct linkage strategy presents two types of adjacency matrix: topological and weighted adjacency matrices. The topological adjacency matrix ignores the fact that different type of relationships may present in such a network between a pair of individuals. It ignores the possibility of multi-link and it shows a connection between u and v if a link present in any one of the interaction network. The topological adjacency matrix $A_{TMN} = \{a_{uv}\}$ of a coupled multiplex network is defined as similar to [47], where

$$a_{uv} = \begin{cases} 1 & if \quad \exists a^{L_i}_{uv} = 1, L_i \in L \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

For example, Fig. 3 shows the topological multiplex network of example graph as shown in Fig. 2 with their corresponding adjacency matrix.

Algorithm 3: Weighted_Coupling($G_i(V_i, E_i, W_i), l$)

Input: $G_i(V_i, E_i, W_i)$, $1 \leq i \leq l$: Social graphs
Output: $G(V, E, W)$: Weighted multiplex social graph

$V\_set \leftarrow \phi$
$Edge\_set \leftarrow \phi$
$Weight \leftarrow \phi$
for *each graph* G_i, $1 \leq i \leq l$ **do**
  for *each edge* $(u, v) \in E_i$ **do**
    if $(u, v) \in Edge\_set$ **then**
      $Weight[u, v] \leftarrow Weight[u, v] + 1$
    else
      $Edge\_set \leftarrow Edge\_set \cup (u, v)$
      $V\_set \leftarrow V\_set \cup \{u, v\}$
      $Weight[u, v] \leftarrow 1$
$Max\_weight \leftarrow max(Weight)$ ▷ Normalize edge weight
for *each edge* $(u, v) \in Edge\_set$ **do**
  $Weight[u, v] \leftarrow \frac{Weight[u, v]}{Max\_weight}$
Return $G(V\_set, Edge\_set, Weight)$

The weighted adjacency matrix of coupled multiplex network considers the possibility of multi-link between a pair of individual. The weighted adjacency matrix $A_{WMN} = \{w(u, v)\}$ is defined as similar to [47], where

$$w(u, v) = \sum_{L_i \in L} a^{L_i}_{uv}; 0 \leq w(u, v) \leq l, \forall u, v \in V \tag{9}$$

For example, Fig. 3 shows the weighted multiplex network of example graph as shown in Fig. 2 with their corresponding weighted adjacency matrix.

Algorithm 4: Weighted_Influence_Coupling(G_i, l, α)

Input: $G_i(V_i, E_i, W_i)$, $1 \le i \le l$: Social graphs, α: Probability vector
Output: $G(V, E, W)$: Weighted influence multiplex graph

```
V_set ← φ
Edge_set ← φ
Weight ← φ
for each graph G_i, 1 ≤ i ≤ l do
    for each edge (u, v) ∈ E_i do
        if (u, v) ∈ Edge_set then
            Weight[u, v] ← Weight[u, v] + α^{L_i}
        else
            Edge_set ← Edge_set ∪ (u, v)
            V_set ← V_set ∪ {u, v}
            Weight[u, v] ← α^{L_i}
Max_weight ← max(Weight)                              ▷ Normalize edge weight
for each edge (u, v) ∈ Edge_set do
    Weight[u, v] ← Weight[u, v] / Max_weight
Return G(V_set, Edge_set, Weight)
```

To construct real dynamics of information propagation in a social network, we need to identify which type of relationship is best suited for information diffusion and which is worst. For this purpose, we uses a probability vector $\alpha = [\alpha^{L_1}, \alpha^{L_2}, \ldots, \alpha^{L_l}]$ to represent importance of each type of relationship. Therefore, weight index $w(u, v)$ of each link (u, v) is calculated as follows.

$$w(u, v) = \sum_{L_i \in L} a_{uv}^{L_i}.\alpha^{L_i}; 0 \le \alpha^{L_i} \le 1, 0 \le \sum_{L_i \in L} \alpha^{L_i} \le l \tag{10}$$

For example, let us consider that probability vector $\alpha = [\alpha^{SN}, \alpha^{RTN}, \alpha^{REN}, \alpha^{MTN}] = [0.2, 1, 0.5, 0.5]$. Fig. 3 shows the weighted-influence multiplex network with probability vector and corresponding weighted matrix. We need to perform normalization of edge weight $w(u, v) \in [0, 1]$ of every individual u and v in weighted and weighted-influence coupling. The overall time complexity of the coupling scheme is $O(\sum_{i=1}^{i=l}(|V_i| + |E_i|))$.

5.4. Multiple influence maximization

In this section, we discuss the IM process for multiple products in a multiplex network achieved from the above discussed coupling strategy.

5.4.1. Generation of product diffusion graph

To maximize the influence spread in the multiplex network for m products simultaneously, MIM2 constructs a social graph corresponding to each product. The edge weight in the social graph of each product follow a probability distribution for information diffusion and all graphs share the same set of users. The influence propagation for each product is independent in the network. The objective of MIM2 is to select k seed nodes from these weighted product diffusion graphs to maximize the overall influence spread of products. The overall complexity of generating m diffusion graph is $O(m(|V| + |E|))$.

5.4.2. Finding non-candidate nodes

The non-candidate nodes are those who less likely to be influential in the network. We find the fixed percentage of non-candidate nodes from the network to trace back to maximum influencing seeds using the reverse graph. To identify non-candidate nodes, first, we remove all the edges with weight less than $w\%$ of the average weight of the network. Second, assign the estimated influence of nodes already selected as seed equal to infinity. Next, assign the estimated influence of remaining nodes equal to the sum of weights of outgoing edges. Finally, we select t least weighted nodes as non-candidate. The overall time complexity of finding non-candidate set in a graph is $O(|E| + |V| \log |V|)$.

Algorithm 5: Finding_Non_Candidates(G_i, t, w)

```
Input: G_i(V_i, E_i, W_i): Social graph, w: Threshold weight percentage, t: Number of non-candidate nodes
Output: Non_Cand: Non-candidate nodes
Sum ← 0
for each edge (u, v) ∈ E_i do
    Sum ← Sum + W_i[u, v]
Average ← Sum / |E|
Threshold ← Average ∗ w
Est_inf ← Initialize estimated influence of each node to 0
for each edge (u, v) ∈ E_i do
    if W_i[u, v] > Threshold then
        Est_inf[u] ← Est_inf[u] + W_i[u, v]
V_Sort ← G_i.V_i in ascending order with respect to estimated influence Est_inf
Non_Cand ← Select first t nodes of V_Sort as non-candidate nodes
Return Non_Cand
```

Algorithm 6: NextSeedA(G^i, *update*, *Acc*, *seed*)

```
Input: G_i(V, E, W^i): Social graph, update: Vector indicated whether influence value is updated, Acc: Vector store
        accumulated influence so far, seed: Current seed set so far
Output: x: Most influential seed
if G^i.update = 1 then
    Return max(Acc)
Nds ← Finding_Non_Candidates(G^i, t, w)
Queue Q
for each node u ∈ Nds do
    Q.push(u)
    u.visited ← 1
Initialize Acc value of each node to 1
while (!Q.empty()) do
    r ← Q.front()
    Q.pop()
    for each edge (v, r) ∈ E do
        x ← v
        if x ∉ seed then
            if !(x.visited = 1) then
                x.visited = 1
                Q.push(x)
            Acc[x] ← Acc[x] + Acc[r] ∗ W^i[v, r]
G^i.update ← 1
Return max(Acc)
```

5.4.3. Identifying most influential user for product P_i

To identify the most influential user for a product P_i, MIM2 selects the corresponding product diffusion graph and performs backward propagation to find the most influential user. First, the algorithm selects non-candidate nodes and then it traverses the graph in reverse order from these nodes using Breadth First Search (BFS) to accumulate the expected influence spread of each node. Finally, the algorithm selects the node with the maximum expected influence as the most influential user for the corresponding product. The time complexity of finding the most influential user over a graph is $O(|E| + |V| \log |V|)$.

5.4.4. Identifying most influential user among all product diffusion graph

The algorithm finds seed nodes by selecting most influential node among all product diffusion graphs iteratively. To find most influential user among all product, it selects most influential user for each product P_i, $1 \leq i \leq m$ by reverse tracing using BFS and compare with each other to find maximal spread node. After finding most influential node, add it

Algorithm 7: NextSeedB(G^i, $update_i$, Acc_i, $seed_i$)

Input: $G^i(V, E, W^i)$, $1 \le i \le m$: Social graphs, $update_i$: Vector indicated whether influence value is updated, Acc_i: Accumulated influence so far, $seed_i$: Current seed set so far
Output: x: Most influential seed among all m-graphs

```
1 x ← Null
2 for each graph G^i, 1 ≤ i ≤ m do
3     x ← max(x, NextSeedA(G^i, update_i, Acc_i, seed_i))        ▷ See Algorithm 6
4 Return x
```

to seed set and delete this node from the corresponding product diffusion graph. This process is continued until k nodes are selected.

5.4.5. Influence estimation

To evaluate the performance of the proposed algorithm, we need to compute the influence spread of selected seed nodes in the coupled multiplex network. We use the same propagation strategy as discussed in Section 4.2 under traditional diffusion models.

6. Algorithm

In this section, we provide a detailed description of each algorithm. The main **Algorithm 1** MIM2 takes four inputs, l social graphs, budget k, number of graphs l, and number of products m. Line 1 initializes seed set S with an empty set. Line 2 assigns *Remaining* to budget k. Line 3 performs coupling on l social graphs and gets a multiplex network G. Line 4 constructs product diffusion graph using normal distribution. Lines 5–10 iteratively finds the most influential node across diffusion graph and adds it to seed set S. Line 11 returns the seed set S as the final output.

The **Algorithm 2** Topological_Coupling takes l influence graphs as input and it returns a coupled multiplex network based on topological structure as output. First of all, lines 1–3 assign an empty set to *V_set*, *Edge_set*, and *Weight*. The **for** loop in lines 4–7 add edge and vertex of individual graphs in *Edge_set* and *V_set*. The **for** loop in lines 8–9 perform coupling based on their topological structure. Line 10 returns a coupled multiplex network.

The **Algorithm 3** Weighted_Coupling takes l influence graphs as input and it returns a coupled multiplex network based on their weighted topological structure as output. Lines 1–3 assign an empty set to *V_set*, *Edge_set*, and *Weight*. Lines 4–11 perform coupling based on their weighted topological structure by considering the different type of relationships. Line 12 finds the maximum possible weight of an edge. Line 13–14 performs normalization of edge weight. Line 15 returns a coupled multiplex network.

The **Algorithm 4** Weighted_Influence_Coupling takes l influence graphs as input and it returns a coupled multiplex network based on their weighted topological structure as output. Lines 1–3 assign an empty set to *V_set*, *Edge_set*, and *Weight*. Lines 4–11 perform coupling based on their weighted influence by considering their different diffusion behavior. Line 12 finds the maximum possible weight of an edge. Line 13–14 performs normalization of edge weight. Line 15 returns a coupled multiplex network.

The **Algorithm 5** Finding_Non_Candidates takes three inputs, a social graph G_i, threshold weight percentage w, and the number of non-candidates t. Line 1 initializes the *Sum* with 0. Lines 2–3 calculate the sum of influence weight of each edge in the graph G_i. Line 4 estimates the average influence of an edge in the graph. Line 5 sets a threshold value for the selection of non-candidate nodes. Line 6 initializes the estimated influence *Est_inf* of each node u to 0. Lines 7–9 compute *Est_inf* for each node u iteratively. Line 10 performs sorting in ascending order based on *Est_inf*. Line 11 selects t non-candidate seed nodes and store in a vector *Non_Cand*. Line 12 returns the *Non_Cand* as an output.

The **Algorithm 6** NextSeedA takes four inputs, a social graph G_i, update status vector *update*, accumulated influence vector *Acc*, and current seed set *seed*. Line 1–2 checks that the influence values of nodes are already updated or not. Line 3 calls Finding_Non_Candidates algorithm and finds non-candidate nodes *Nds*. Line 4 maintains a queue Q. Lines 5–7 marks non-candidate nodes as visited and push *Nds* nodes into Q. Line 8 initializes accumulated influence *Acc* of each node to 1. Line 9–19 updates *Acc* iteratively for each node. Line 10–11 pop the node from the front and store it to r, if queue Q is not empty. Lines 12–18 updates *Acc* iteratively based on each edge. Line 19 marks the graph as updated. Line 20 returns maximum accumulated influence node as most influential node in the graph.

The **Algorithm 7** NextSeedB takes four inputs, m social graphs, update status vector $update_i$, accumulated influence vector Acc_i, and current seed set $seed_i$ for each graph G_i. Line 1 initializes most influential node x to null. Lines 2–3 find the most influential user across m social graphs and store it to x. Line 4 returns x as most influential user.

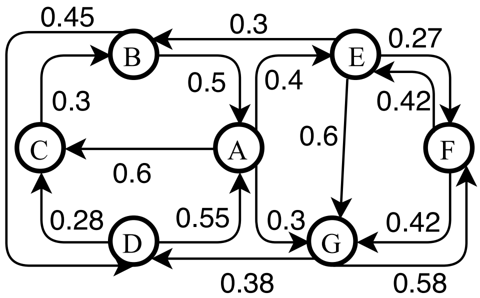


Fig. 4. A running example.

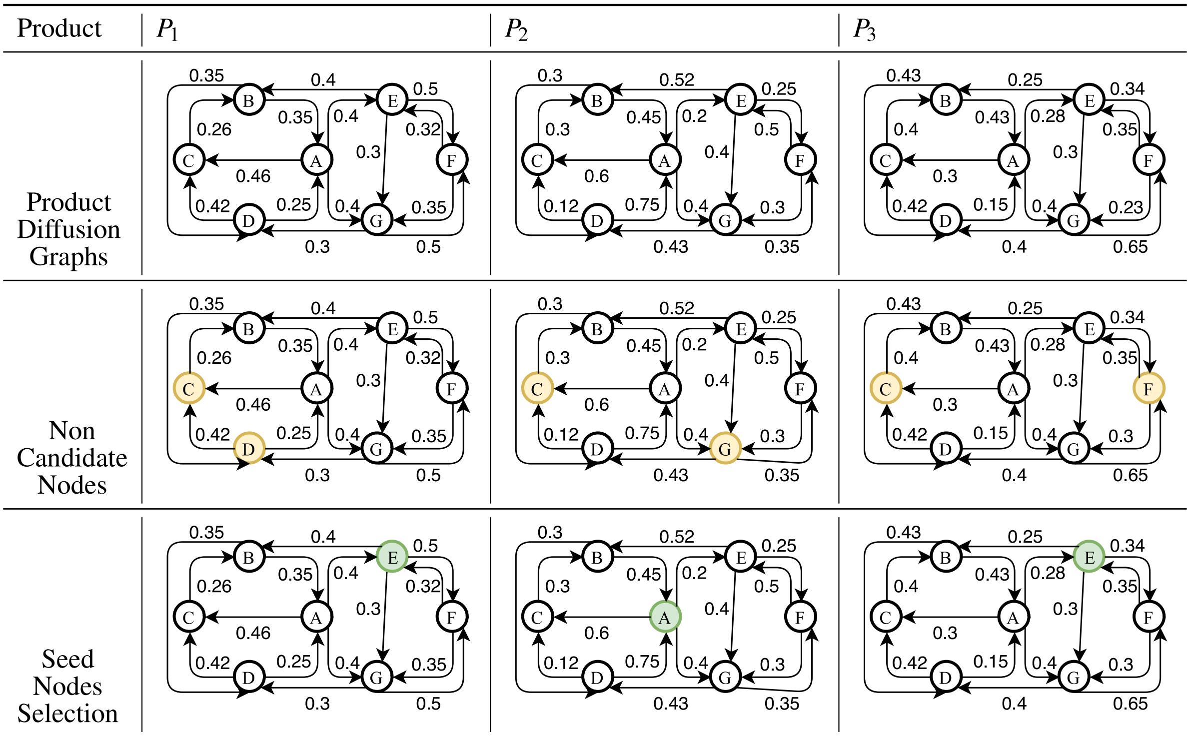


Fig. 5. The working of seed selection process in MIM2 framework.

6.1. Applying the algorithm

In this section, we present a running example to explain the working of the proposed algorithm for MIM2 problem. Let consider a coupled multiplex network and $m = 3$, which is constructed using the weighted-influence coupling scheme and shown in Fig. 4. In the seed selection process of MIM2, first, we construct the product diffusion graphs for each product P_i. Let assume that the product diffusion graphs are constructed using normal distribution and given in Fig. 5. Next, MIM2 selects non-candidate nodes in each product diffusion graph G^i. Each edge (u, v) in G^i has edge weight $W^i[u, v] \leftarrow W_i[u, v] * W_D^{P_i}[u, v]$, where $W_i[u, v]$ and $W_D^{P_i}[u, v]$ denote edge weight of edge (u, v) in the coupled multiplex graph and normal distribution edge in diffusion graph of product P_i. To find non-candidate nodes, we consider $w = 80\%$, $t = 2$ and calculates expected influence *Average* for P_1 as $Average \leftarrow Sum/|E|$, i.e. $Average \leftarrow ((0.4+0.4+0.46)+(0.35+0.35)+(0.26)+(0.25+0.42)+(0.5+0.4+0.3)+(0.32+0.35)+(0.5+0.3))/15 = 0.37$. Before calculating estimated influence $Est\_Inf[u]$ of a node u, we compute threshold value $Threshold \leftarrow 0.37 * 0.8 = 0.29$. Now, we compute $Est\_Inf[u]$ of each node u. The Est_Inf is calculated as $Est\_Inf \leftarrow [0.4 + 0.4 + 0.46, 0.35 + 0.35, 0, 0.42, 0.5 + 0.4 + 0.3, 0.32 + 0.35, 0.5 + 0.3] = [1.26, 0.7, 0, 0.42, 1.2, 0.67, 0.8]$. Next, we select least-$t$ nodes based on Est_Inf value as non-candidate nodes N_C, i.e., $N_C \leftarrow \{C, D\}$ for product diffusion graph G^1. Let non-candidate nodes for graph G^2 and G^3 are $\{C, G\}$ and $\{C, F\}$ respectively. The yellow colored nodes represent non-candidate nodes as in Fig. 5.

Table 3
Computation of accumulated influence *Acc* based on algorithm NextSeedA for product diffusion graph G^1.

Queue Q	$\langle Acc, visited\rangle$ value						
	A	B	C	D	E	F	G
ϕ	1,0	1,0	1,0	1,0	1,0	1,0	1,0
$\{C, D\}$	1,0	1,0	1,1	1,1	1,0	1,0	1,0
$\{D, A\}$	1.46,1	1,0	1,1	1.42,1	1,0	1,0	1,0
$\{A, B, G\}$	1.46,1	1.49,1	1,1	1.42,1	1,0	1,0	1.42,1
$\{B, G\}$	1.46,1	2,1	1,1	1.78,1	1,0	1,0	1.42,1
$\{G, E\}$	1.46,1	2,1	1.52,1	1.78,1	1.8,1	1,0	1.42,1
$\{E, F\}$	2.02,1	2,1	1.52,1	1.78,1	2.22,1	1.49,1	1.42,1
$\{F\}$	2.9,1	2,1	1.52,1	1.78,1	2.22,1	2.2,1	1.42,1
ϕ	2.9,1	2,1	1.52,1	1.78,1	3.32,1	2.2,1	2.52,1

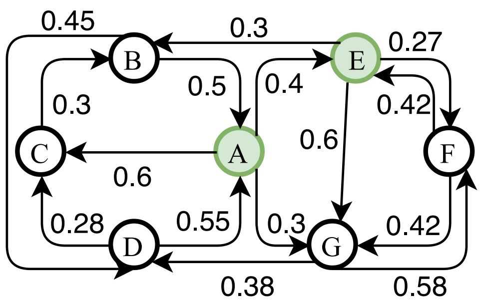


Fig. 6. The final seed nodes.

Next, MIM2 selects seed by reverse tracing using BFS and assume budget $k = 3$. To select the most influential node in graph G^1, initially, queue Q has N_C nodes, i.e. $Q \leftarrow \{C, D\}$ and marks these node as visited. The accumulated influence *Acc* for each node is 1 at start of reverse tracing. The accumulated influence $Acc \leftarrow [2.9, 2, 1.52, 1.78, 3.32, 2.2, 2.52]$ of G^1 is computed based on algorithm NextSeedA as shown in Table 3. Therefore, most influential user in G^1 is E. We assume that most influential user for graph G^2 and G^3 are A and E respectively. Using algorithm NextSeedB, MIM2 selects most influential user among all products is E for P_1. Therefore, E is added to seed set S and removes from G^1. Similarly, remaining seed nodes are selected. Node E and A are selected from graph G^3 and G^2 respectively as seed. The green colored nodes represent seed nodes in the respected diffusion graph, as shown in Fig. 5. Finally, MIM2 procedure combines the all seed set S_i for G_i. The overall seed set $S \leftarrow \bigcup_{i=1}^{i=m} = \{E\} \cup \{A\} \cup \{E\} = \{A, E\}$, $\sum_{i=1}^{i=m} |S_i^*| = k = 3$ and $|S| = 2 \leq k$. Fig. 6 shows the final seed nodes in green color under the MIM2 framework.

6.2. Complexity analysis

In this section, we analyze the worst-case time complexity of proposed algorithm MIM2. Lines 1–2 take $O(1)$ basic operations to initialize S and *Remaining*. Line 3 performs coupling procedure in $O(\sum_{i=1}^{i=l}(|V_i| + |E_i|))$ basic operations, i.e. $O(|V| + |E|)$, where V and E denotes the node and edge set of a multiplex network (also known as coupled network). Line 4 constructs m product diffusion graph in $O(m(|V| + |E|))$ basic operations. Lines 5–10 performs seed selection process iteratively. First, it selects non-candidate nodes using $O(\sum_{i=1}^{i=m}(|E| + |V| \log |V|))$ basic operations. Then, it finds overall seed set from MIM perspective and take $O(k(|E| + |V| \log |V|))$ basic operations. So overall complexity of lines 5–10 is $O((m + k)(|E| + |V| \log |V|))$. Therefore, the worst-case time complexity of MIM2 algorithm is $O((l + m)(M + N) + (k + m)(M + N \log N))$, where $|V| = N$ and $|E| = M$.

6.3. Compared with the state-of-the-art algorithms

We categorize the classical IM problem presented in [8] into four classes: IM, IM2, MIM, and MIM2 problem. We categorize existing state-of-the-art algorithms into these four problems and compare the characteristics of these algorithms with the proposed algorithm. Table 4 provides a theoretical analysis of above-discussed IM approaches. Column 1 categorizes the existing approaches in four classes: IM, IM2, MIM, and MIM2. The column *Algorithm* gives the name of the algorithm with reference. The column *Time Complexity* gives the worst-case time complexity of the algorithm. The column *Approximation* states the approximation ratios of the algorithm (N.A. indicates no approximation guaranteed). The column *Problem Solving Perspective* gives the framework used in the algorithm. The column *State-of-the-art Algorithm* gives the name of the state-of-the-art methods. The column *Base Algorithm* gives the name of the base algorithm for the corresponding algorithm. Table 5 compare the characteristic such as diffusion models applicable, category, and network, etc., of MIM2 algorithm with the existing algorithms.

Table 4
Comparison of the characteristics of the existing IM algorithms with the proposed algorithm.

	Algorithm	Time complexity	Approximation	Problem solving perspective	State-of-the-art algorithms	Base algorithm
IM	Greedy [8]	$O(kNMI)$	$1-1/e-\varepsilon$	Spread Simulation	MaxDegree, Central & Random	–
	Knapsack Greedy [9]	$O(N^5)$	$1-1/e-\varepsilon$	Spread Simulation	–	Greedy
	SP1M [15]	$O(kNM)$	$1-1/e$	Influence Path	Degree, PageRank & Closeness	–
	CELF [11]	$O(kNMI)$	$1-1/e-\varepsilon$	Sub-modularity	Greedy	Greedy
	Degree Discount [13]	$O(k\log N+M)$	N.A.	Heuristic based	CELF, Greedy & Random	High Degree
	NewGreedy [13]	$O(kIM)$	$1-1/e-\varepsilon(r)$	Snapshots	CELF, Greedy & Random	High Degree
	TW Greedy [10]	$O(kNMI)$	$1-1/e-\varepsilon$	Spread Simulation	SCG, KKG & High Degree	Greedy
	MIA / PMIA [16]	$O(Nt_{i\theta}+kn_{o\theta}n_{i\theta}(n_{i\theta}+\log N))$	$1-1/e$	Influence Path	Greedy, Random, DD & PageRank	SP1M
	LDAG [26]	$O(Nt_{\theta}+kn_{\theta}m_{\theta}(m_{\theta}+\log N))$	N.A.	Score Estimation	Greedy, SPIN, DD & PageRank	–
	CGA [18]	$O(M+IM_C(N(Z-C)+k(C+N_C)))$	$1-e^{-1/(1+\delta_C)}$	Community Based	DD, MG & Random	–
	CELF++ [12]	$O(kNMI)$	$1-1/e-\varepsilon$	Sub-modularity	CELF	CELF
	Diffusion Degree [14]	$O(N+M)$	N.A.	Centrality Based	DD & High Degree	High Degree
	SIMPATH [27]	$O(klNP_{\theta})$	N.A.	Score Estimation	High Degree, CELF & PageRank	LDAG
	IRIE [28]	$O(k(n_{o\theta}k+M))$	N.A.	Score Estimation	Greedy & PMIA	–
	IPA [17]	$O(\frac{NO_v n_{vu}}{c}+k^2(\frac{O_v n_{vu}}{c}+(c-1)))$	N.A.	Influence Path	Greedy, DD & Random	PMIA
	StaticGreedy [21]	$O(\frac{kMN^2\log\binom{N}{k}}{\varepsilon^2})$	$1-1/e-\varepsilon$	Snapshots	CELF, SP1M, DD & High Degree	PMIA
	PRUNEDMC [22]	$O(\frac{kMN^2\log\binom{N}{k}}{\varepsilon^2})$	$1-1/e-\varepsilon$	Snapshots	IRIE, Random, PMIA & Degree	Greedy
	TIM [23]	$O(\frac{k(M+N)\log N}{\varepsilon^2})$	$1-1/e-\varepsilon$	Reverse Reachability	CELF++, IRIE & SIMPATH	–
	DPSO [48]	$O(k^2\log kn\bar{D}^2)$	N.A.	Swam Optimization	Degree, CELF++ & SAEDV	–
IM2	BP-Greedy [49]	–	$1-1/e$	Spread Estimation	Betweenness, Degree, & Closeness	Greedy
	MPMN-CELF++ [50]	$O(kNMI)$	N.A.	Spread Simulation	SIMPATH & CELF++	CELF++
	MPMN-SIMPATH [50]	$O(klNP_{\theta})$	N.A.	Influence Ranking	SIMPATH & CELF++	SIMPATH
	ASMTC [51]	$O(\|V^s\|^2+\|V^s\|)$	N.A.	Reverse Reachability	–	–
	SeedSelection-M [52]	–	N.A.	Rank Refinement	Degree, K-Shell & VoteRank	–
	LCI [32]	$O((N+M)N.d)$	N.A.	Sub-modularity	Greedy	Greedy
MIM	MIM-Greedy [33]	$O(kmNMI)$	$1-1/e$	Spread Simulation	MaxDegree, Init-First & Random	Greedy
MIM2	**MIM2**	$O((l+m)(M+N)+(k+m)(M+N\log N))$	N.A.	Heuristic based	MaxDegree, DD, & MIM-Greedy	C2IM

7. Experimental evaluation

In this section, we explain the experimental setup and the results of the conducted experiments. We evaluate the proposed algorithm MIM2 with the state-of-the-art seeding strategies from the following aspects: (a) comparison of the coupling schemes; (b) advantage of coupled network; (c) advantage of MIM2 framework; (d) the effectiveness in terms of influence spread; (b) the efficiency in terms of running time.

7.1. Experimental setup

To evaluate the proposed seeding strategy, we compare MIM2 with best suited existing algorithm under different coupling scheme. To perform experiments, we employ real-world networks. The propagation probabilities and activation probabilities are produced from well-known distributions, as described later in the section.

7.1.1. Network structure

In order to perform experiment, we use two datasets: Higgs Twitter networks[3] [54] and co-author networks[4] [32,55]. Higgs dataset extracted from Twitter network based on user activities on elusive Higgs boson discovery between 1st and 7th July 2012. The Twitter network is follower–follow relationship network (FN) and it consists three types of interaction networks: Retweet network (RTN), Reply network (REN) and Mention network (MEN). The authors of [54] perform experiments on different periods and divide network based on these experiment time period. The experiment periods are: before 1 PM GMT on 2nd July (Period I), after 2nd July and before the announcement on 4th July (Period II), and after announcement to 7th July (Period III). The statistical information of Higgs dataset is given in Tables 6 and 7.

3 http://snap.stanford.edu/data/higgs-twitter.html.

4 https://www.cise.ufl.edu/research/OptimaNetSci/tools/id_inter.html.

Table 5
Comparison of the characteristics of the existing IM algorithms with the proposed algorithm.

	Algorithm	Diffusion model				Simulation	Heuristic	Meta-heuristic	Sketch	Network	
		Linear threshold	Independent cascade	Triggering	Continuous time-aware					Single	Multiple
IM	Greedy [8]	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗
	Knapsack Greedy [9]	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗
	SP1M [15]	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗
	CELF [11]	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗
	Degree Discount [13]	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗
	NewGreedy [13]	✗	✓	✗	✗	✗	✗	✗	✓	✓	✗
	TW Greedy [10]	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗
	MIA / PMIA [16]	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗
	LDAG [26]	✓	✗	✗	✗	✗	✓	✗	✗	✓	✗
	CGA [18]	✗	✓	✗	✗	✓	✗	✗	✗	✓	✗
	CELF++ [12]	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗
	Diffusion Degree [14]	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗
	SIMPATH [27]	✓	✗	✗	✗	✗	✓	✗	✗	✓	✗
	IRIE [28]	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗
	IPA [17]	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗
	StaticGreedy [21]	✗	✓	✗	✗	✗	✗	✗	✓	✓	✗
	PRUNEDMC [22]	✗	✓	✗	✗	✗	✗	✗	✓	✓	✗
	TIM [23]	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗
	ACO [53]	✓	✓	✓	✓	✗	✗	✓	✗	✓	✗
	DPSO [48]	✓	✓	✓	✓	✗	✗	✓	✗	✓	✗
IM2	BP-Greedy [49]	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓
	MPMN-CELF++ [50]	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓
	MPMN-SIMPATH [50]	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
	ASMTC [51]	✗	✓	✗	✗	✗	✗	✗	✓	✗	✓
	SeedSelection-M [52]	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓
	LCI [32]	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓
MIM	MIM-Greedy [33]	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗
MIM2	**MIM2**	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓

Table 6
Statistical information of Higgs dataset.

	FN	RTN	RE	MEN
Nodes $\lvert V_i \rvert$	456 626	256 491	38 918	116 408
Edges $\lvert E_i \rvert$	14 855 842	328 132	32 523	150 818

Table 7
Statistical information of follower network on Higgs dataset over different period.

	Period I	Period II	Period III
Nodes $\lvert V_i \rvert$	6 061	87 457	247 440
Edges $\lvert E_i \rvert$	7 712	169 526	378 243

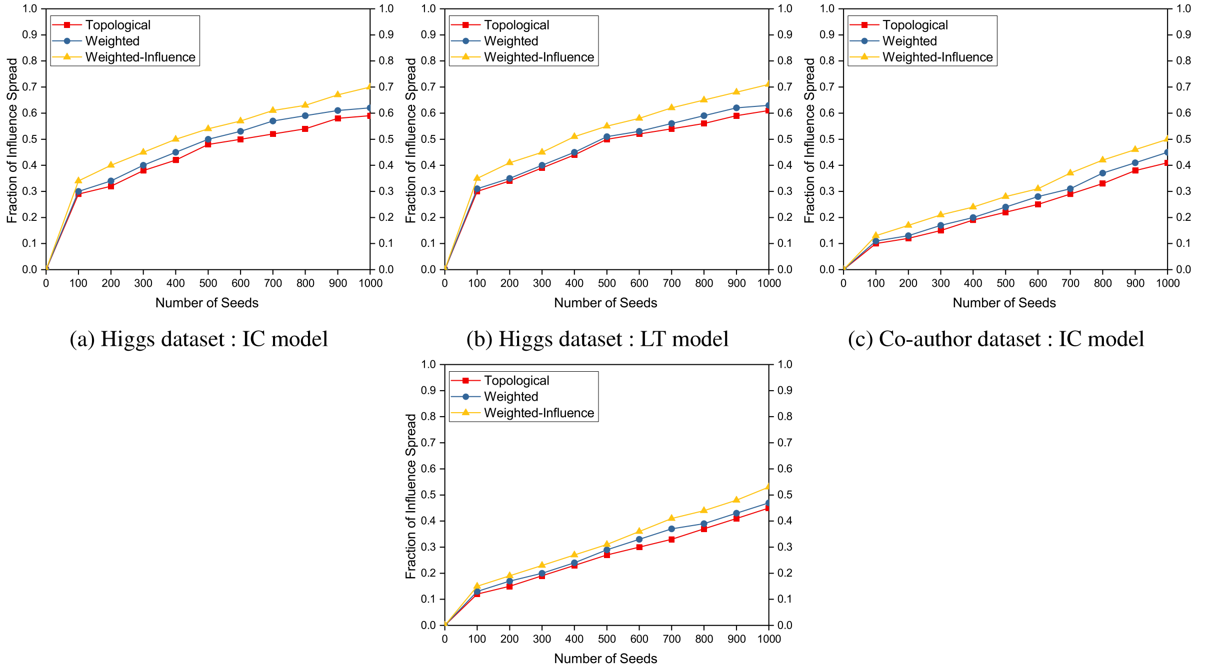
Table 8
Statistical information of co-author dataset.

	CM	NS	HT
Nodes $\lvert V_i \rvert$	40 420	1 588	6 360
Edges $\lvert E_i \rvert$	175 692	2 742	15 751

The co-author dataset consists of three citation network in the field of Condensed Matter(CM), Network Science(NS), and High-Energy Theory(HT). The statistical information of co-author dataset is given in Table 8. To identify overlapping users in co-author dataset, we use authors name for matching. The number of overlapping users are 2860, 90, and 517 for network pairs CM–HT, HT–NS, and CM–NS respectively.

7.1.2. Probability distribution

The Higgs dataset only provides topology information so we need to assign edge weight or propagation probability of each edge (u, v). In IC model, the propagation probability follows uniform discrete distribution over {0.1,0.01,0.001}. In LT, the propagation probability of (u, v) is assigned to $\frac{1}{indegree(v)}$. In co-author dataset edge weights are given. The



(a) Higgs dataset : IC model (b) Higgs dataset : LT model (c) Co-author dataset : IC model (d) Co-author dataset : LT model

Fig. 7. Comparison of the coupling schemes over MIM2 influence spread for $m = 1$.

propagation probability of (u, v) of product diffusion graph for both datasets follows a normal distribution. The assignment of activation probability of each node u follows a uniform distribution.

7.1.3. Seeding strategies to compare

We adopt following state-of-the-art algorithms to evaluate the performance of the proposed algorithm under traditional diffusion models.

- **Random.** This method selects k seed nodes randomly [8].
- **MaxDegree.** This method selects k highest out-degree nodes as seed nodes [8].
- **Degree Discount.** This approach is based on the concept that after a node is selected as a seed node then it will no longer be influenced by its neighbors [13].
- **MIM-Greedy.** This approach performs IM based on a greedy approach for multiple products simultaneously [33]. It takes the assumption that influence of a node is spread up to 6 hope.
- **MIM2.** This is our proposed heuristic algorithm. It selects seed by reverse tracing from non-candidate nodes across multiple networks for multiple products simultaneously.

7.2. Comparison of coupling schemes

We evaluate the performance of coupling schemes on the influence spread of MIM2 algorithm. Fig. 7 shows that the weighted-influence coupling scheme activates more users than other two coupling schemes under traditional diffusion models for both Twitter and co-author networks. This is because of weighted-influence coupling considers the importance of different types of relationships for information diffusion along with the topology structure of the network.

7.3. Advantages of using IM2

In order to understand the benefit of coupled multiplex networks under IM2 framework, we compare the influence spread of proposed algorithm for different seed size over coupled networks. Fig. 8 shows the effect of avoiding one type of relationship in Higgs Twitter networks and co-author networks under a traditional diffusion model. In all three periods of Higgs dataset under both diffusion models, the avoidance of RTN edges causes the most significant difference in influence spread. The influence spread significantly deteriorate when RTN edges not considered in the influence

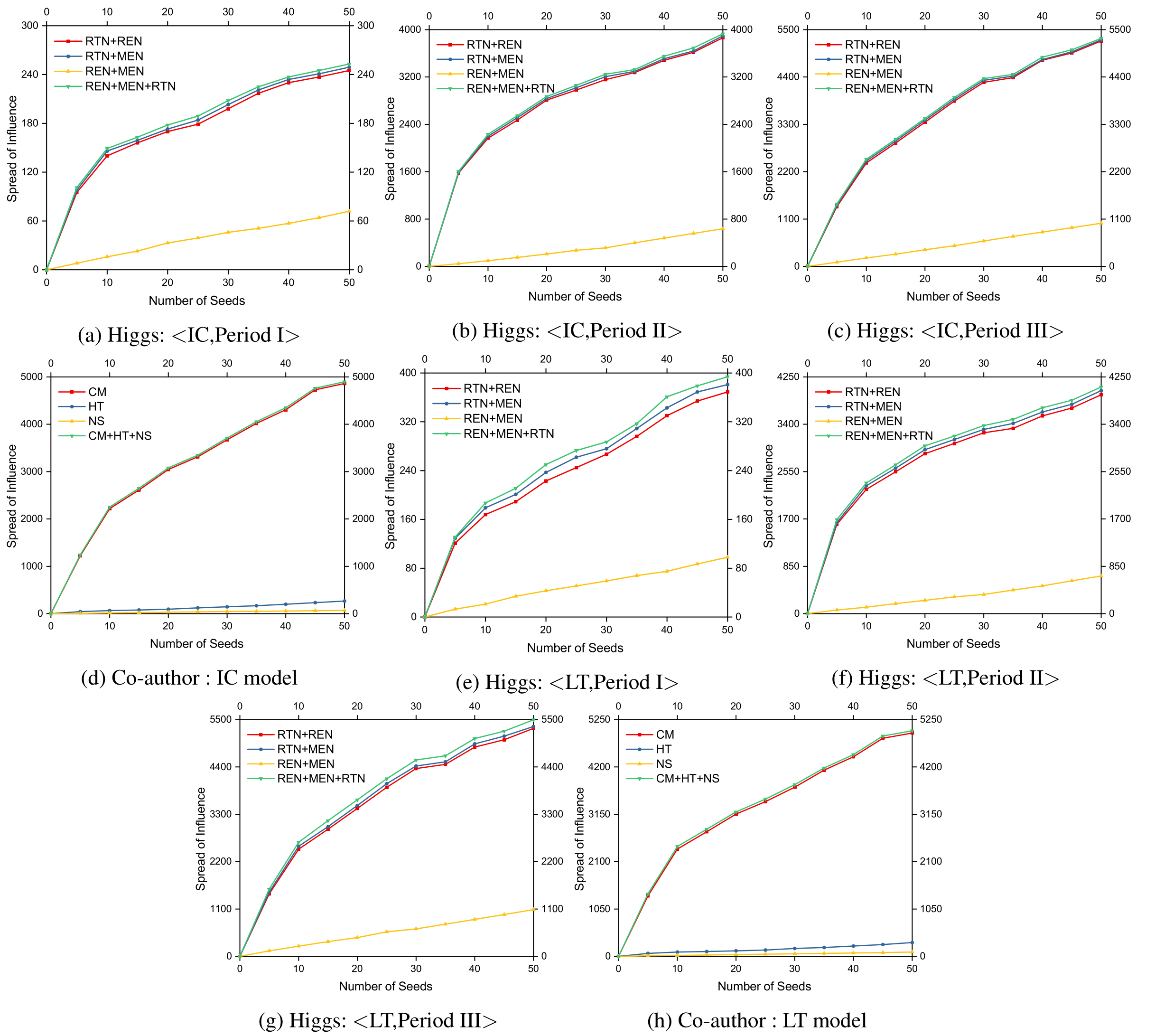


Fig. 8. Comparison of the influence spread over IM2 framework for $m = 1$ under traditional diffusion models.

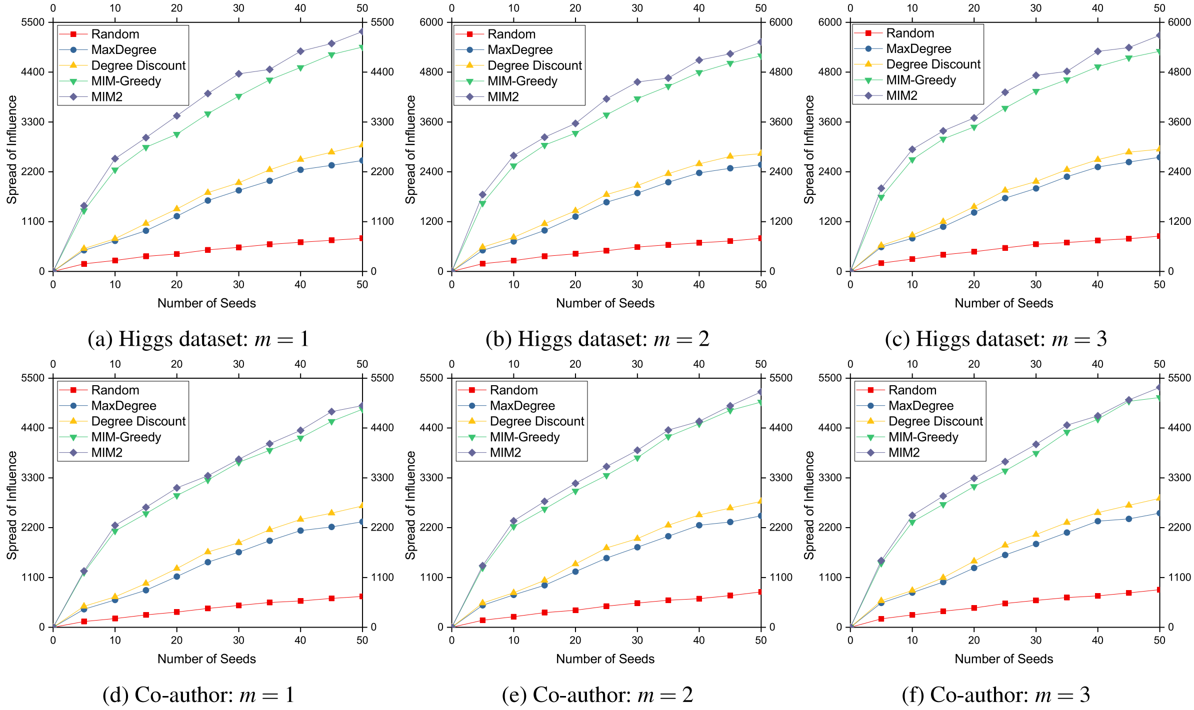
Table 9
The influence spread comparison of algorithm under IM2 framework for Higgs Twitter dataset at $k = 50$.

		Influence spread			
		RTN+REN	RTN+MEN	REN+MEN	RTN+REN+MEN
IC	Period I	245	249	72	253
	Period II	3864	3896	637	3929
	Period III	5238	5262	1003	5298
LT	Period I	369	381	98	394
	Period II	3929	4004	677	4069
	Period III	5298	5342	1085	5498

diffusion process. Fig. 8 also illustrates the effect of the coupled network in co-author networks. The influence spread is very less when we consider only HT or only NS network. Tables 9 and 10 provide the influence spread of MIM2 algorithm in coupled multiplex networks under IM2 framework at $k = 50$ for Higgs Twitter and co-author datasets respectively.

Table 10
The influence spread comparison of algorithm under IM2 framework for co-author dataset at $k = 50$.

Diffusion model	Influence spread			
	CM	HT	NS	CM+HT+NS
IC	4762	265	68	4857
LT	4952	305	94	5063



(a) Higgs dataset: $m = 1$ (b) Higgs dataset: $m = 2$ (c) Higgs dataset: $m = 3$

(d) Co-author: $m = 1$ (e) Co-author: $m = 2$ (f) Co-author: $m = 3$

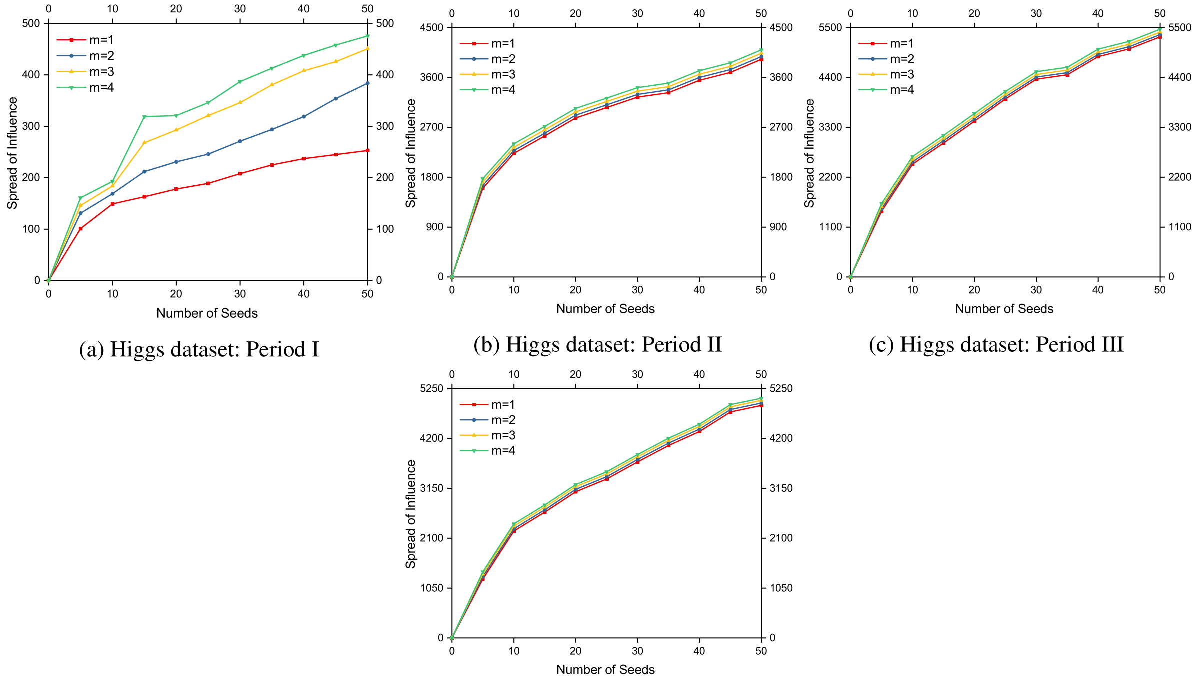
Fig. 9. Comparison of the influence spread over MIM framework for different number of product m under IC model.

Table 11
The influence spread comparison of compared algorithms under MIM framework for different m at $k = 50$.

Dataset	m	Influence spread				
		Random	MaxDegree	Degree Discount	MIM-Greedy	MIM2
Twitter	1	731	2451	2789	4954	5298
	2	798	2571	2839	5194	5528
	3	853	2751	2945	5304	5688
Co-author	1	681	2331	2679	4814	4897
	2	781	2461	2779	4974	5197
	3	831	2521	2849	5074	5297

7.4. Advantages of using MIM

In our experiments, we assume number of products $m = 1, 2, 3$ and seed size $k = 5, 10, \ldots, 50$. Fig. 9 shows the influence spread of compared algorithms for different size seed set and for different value of m under IC model. We can see that MIM2 algorithm outperforms other algorithms under MIM framework for the different number of products. This is because of Random, MaxDegree, and Degree Discount heuristic methods are only dependent on network topology and edge weight. MIM-Greedy is based on the influence matrix which considers influence up to six hope. Our MIM2 algorithm estimates the diffusion degree of each node by reverse tracing the whole network. Figs. 9a to 9c illustrate that the influence spread of algorithms are increases with the increase of the number of products in Higgs dataset. Similarly, Figs. 9d to 9f compares the influence spread of algorithm for different value of m. From these experiments we can observe that the influence spread of MIM2 algorithm is increases with increment in number of products. This influence spread increment is highly evident until k reaches a certain threshold. This is because of the



(a) Higgs dataset: Period I

(b) Higgs dataset: Period II

(c) Higgs dataset: Period III

(d) Co-author dataset

Fig. 10. Comparison of the influence spread of MIM2 framework for different number of product m under IC model.

remaining nodes have less social source, in-neighbors and out-neighbors. Table 11 gives the influence spread of compared algorithm under IC diffusion model for MIM framework at $k = 50$.

7.5. Advantages of using MIM2

To understand the benefit of proposed MIM2 framework, we perform experiment on influence spread and running time of proposed algorithm for different diffusion graphs under traditional diffusion models.

7.5.1. Influence spread

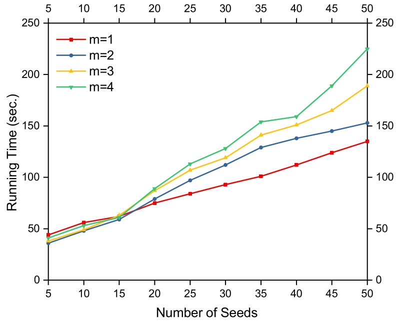
Fig. 10 shows the influence spread of MIM2 algorithm for different m values under IC model. We can see that the influence spread increases with growth in seed size. It is also evident that the influence spread of MIM2 algorithm increases with increment in the number of diffusion graphs k. These experimental results show that our influence model and MIM2 framework are full of effectiveness.

7.5.2. Running time

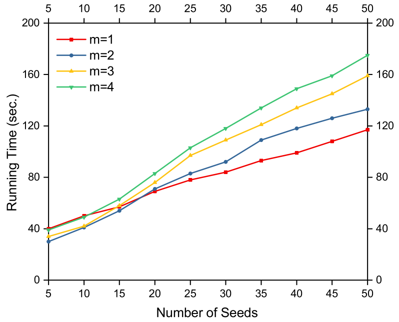
Apart from the influence spread comparison, we also compare the running time of proposed algorithm under MIM2 framework for different m ranges from 1 to 4. Fig. 11 shows the running time of our algorithm for both datasets. We can see that the running time increases with the growth of seed set k as well as the number of product diffusion graphs m. The running time of selecting seed under MIM2 framework takes very few minutes. This shows that MIM2 algorithm has good performance. Therefore, we can say that MIM2 algorithm has good quality solution and efficiency.

8. Conclusion and future directions

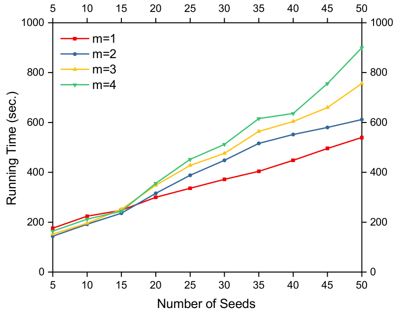
In this paper, we present a *multiple influence maximization across multiple networks* (MIM2) problem to tackle the scenario that multiple products can be promoted in a network with different channel of interactions. To effectively solve this problem, we presented MIM2 algorithm. First, we introduced a coupling scheme to map multiple interaction networks into a single multiplex network. Then, we presented a procedure to find non-candidate nodes for backward tracing. Next, we constructed product diffusion graphs for each product. Finally, we provided a seed selection process across product diffusion graphs. Exhaustive experiments provided new insights into the MIM2 problem and the benefits of MIM2 framework. This work opens to several future work directions such as the investigation of MIM2 problem in multiplex networks with heterogeneous diffusion models with some constraints like competitiveness, budget, location, time, etc.



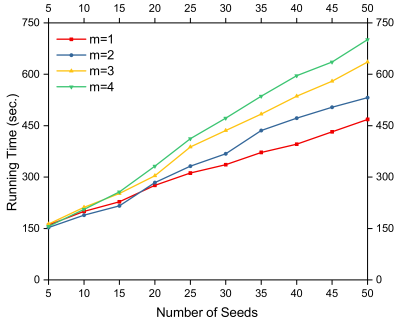
(a) Higgs dataset: IC model



(b) Higgs dataset: LT model



(c) Higgs dataset: IC model



(d) Co-author: LT model

Fig. 11. Comparison of the running time of MIM2 framework for different number of product m.

9. Compliance with Ethical Standards

The article does not contain any studies with human or animal subjects. This article present an algorithm MIM2.

Conflict of interest

The authors declare no conflicts of interest.

References

[1] J. Goldenberg, B. Libai, E. Muller, Talk of the network: A complex systems look at the underlying process of word-of-mouth, Mark. Lett. 12 (3) (2001) 211–223, http://dx.doi.org/10.1023/A:1011122126881.
[2] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '01, ACM, New York, NY, USA, 2001, pp. 57–66, http://dx.doi.org/10.1145/502512.502525.
[3] P. Wu, L. Pan, Scalable influence blocking maximization in social networks under competitive independent cascade models, Comput. Netw. 123 (2017) 38–50, http://dx.doi.org/10.1016/j.comnet.2017.05.004.
[4] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: Proceedings of the 2010 IEEE International Conference on Data Mining, in: ICDM '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 88–97, http://dx.doi.org/10.1109/ICDM.2010.118.
[5] Y.-W. Teng, C.-H. Tai, P.S. Yu, M.-S. Chen, Revenue maximization on the multi-grade product, in: Proceedings of the 2018 SIAM International Conference on Data Mining, 2018, pp. 576–584, http://dx.doi.org/10.1137/1.9781611975321.65.
[6] M. Ye, X. Liu, W.-C. Lee, Exploring social influence for recommendation: A generative model approach, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR '12, ACM, New York, NY, USA, 2012, pp. 671–680, http://dx.doi.org/10.1145/2348283.2348373.
[7] J. Leskovec, L.A. Adamic, B.A. Huberman, The dynamics of viral marketing, ACM Trans. Web 1 (1) (2007) http://dx.doi.org/10.1145/1232722.1232727.
[8] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '03, ACM, New York, NY, USA, 2003, pp. 137–146, http://dx.doi.org/10.1145/956750.956769.
[9] M. Sviridenko, A note on maximizing a submodular set function subject to a knapsack constraint, Oper. Res. Lett. 32 (1) (2004) 41–43, http://dx.doi.org/10.1016/S0167-6377(03)00062-2.
[10] Y. Wang, X. Feng, A potential-based node selection strategy for influence maximization in a social network, in: R. Huang, Q. Yang, J. Pei, J. Gama, X. Meng, X. Li (Eds.), Advanced Data Mining and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 350–361.

[11] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '07, ACM, New York, NY, USA, 2007, pp. 420–429, http://dx.doi.org/10.1145/1281192.1281239.

[12] A. Goyal, W. Lu, L.V. Lakshmanan, Celf++: Optimizing the greedy algorithm for influence maximization in social networks, in: Proceedings of the 20th International Conference Companion on World Wide Web, in: WWW '11, ACM, New York, NY, USA, 2011, pp. 47–48, http://dx.doi.org/10.1145/1963192.1963217.

[13] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '09, ACM, New York, NY, USA, 2009, pp. 199–208, http://dx.doi.org/10.1145/1557019.1557047.

[14] S. Kundu, C.A. Murthy, S.K. Pal, A new centrality measure for influence maximization in social networks, in: S.O. Kuznetsov, D.P. Mandal, M.K. Kundu, S.K. Pal (Eds.), Pattern Recognition and Machine Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 242–247.

[15] M. Kimura, K. Saito, Tractable models for information diffusion in social networks, in: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Knowledge Discovery in Databases: PKDD 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 259–271.

[16] W. Chen, C. Wang, Y. Wang, 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, in: KDD '10, ACM, New York, NY, USA, 2010, pp. 1029–1038, http://dx.doi.org/10.1145/1835804.1835934.

[17] J. Kim, S.K. Kim, H. Yu, Scalable and parallelizable processing of influence maximization for large-scale social networks? in: 2013 IEEE 29th International Conference on Data Engineering (ICDE), 2013, pp. 266–277, http://dx.doi.org/10.1109/ICDE.2013.6544831.

[18] Y. Wang, G. Cong, G. Song, K. Xie, Community-based greedy algorithm for mining top-k influential nodes in mobile social networks, in: KDD, 2010.

[19] H. Li, S.S. Bhowmick, A. Sun, J. Cui, Conformity-aware influence maximization in online social networks, VLDB J. 24 (1) (2015) 117–141, http://dx.doi.org/10.1007/s00778-014-0366-x.

[20] S.S. Singh, K. Singh, A. Kumar, B. Biswas, Coim: Community-based influence maximization in social networks, in: A.K. Luhach, D. Singh, P.-A. Hsiung, K.B.G. Hawari, P. Lingras, P.K. Singh (Eds.), Advanced Informatics for Computing Research, Springer Singapore, Singapore, 2019, pp. 440–453.

[21] S. Cheng, H. Shen, J. Huang, G. Zhang, X. Cheng, Staticgreedy: Solving the scalability-accuracy dilemma in influence maximization, in: Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, in: CIKM '13, ACM, New York, NY, USA, 2013, pp. 509–518, http://dx.doi.org/10.1145/2505515.2505541, URL http://doi.acm.org/10.1145/2505515.2505541.

[22] N. Ohsaka, T. Akiba, Y. Yoshida, K.-I. Kawarabayashi, Fast and accurate influence maximization on large networks with pruned monte-carlo simulations, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, in: AAAI'14, AAAI Press, 2014, pp. 138–144, URL http://dl.acm.org/citation.cfm?id=2893873.2893897.

[23] Y. Tang, X. Xiao, Y. Shi, Influence maximization: near-optimal time complexity meets practical efficiency, in: SIGMOD Conference, 2014.

[24] D. Varshney, S. Kumar, V. Gupta, Predicting information diffusion probabilities in social networks: A bayesian networks based approach, Knowl.-Based Syst. 133 (2017) 66–76, http://dx.doi.org/10.1016/j.knosys.2017.07.003.

[25] D. Varshney, S. Kumar, V. Gupta, Modeling information diffusion in social networks using latent topic information, in: D.-S. Huang, V. Bevilacqua, P. Premaratne (Eds.), Intelligent Computing Theory, Springer International Publishing, Cham, 2014, pp. 137–148.

[26] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: 2010 IEEE International Conference on Data Mining, 2010, pp. 88–97, http://dx.doi.org/10.1109/ICDM.2010.118.

[27] A. Goyal, W. Lu, L.V.S. Lakshmanan, Simpath: An efficient algorithm for influence maximization under the linear threshold model, in: Proceedings of the 2011 IEEE 11th International Conference on Data Mining, in: ICDM '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 211–220, http://dx.doi.org/10.1109/ICDM.2011.132.

[28] K. Jung, W. Heo, W. Chen, Irie: Scalable and robust influence maximization in social networks, in: Proceedings of the 2012 IEEE 12th International Conference on Data Mining, in: ICDM '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 918–923, http://dx.doi.org/10.1109/ICDM.2012.79.

[29] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, J. Han, Event-based social networks: Linking the online and offline social worlds, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '12, ACM, New York, NY, USA, 2012, pp. 1032–1040, http://dx.doi.org/10.1145/2339530.2339693.

[30] O. Yagan, D. Qian, J. Zhang, D. Cochran, Information diffusion in overlaying social-physical networks, in: 2012 46th Annual Conference on Information Sciences and Systems (CISS), 2012, pp. 1–6.

[31] Y. Shen, T.N. Dinh, H. Zhang, M.T. Thai, Interest-matching information propagation in multiple online social networks, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, in: CIKM '12, ACM, New York, NY, USA, 2012, pp. 1824–1828, http://dx.doi.org/10.1145/2396761.2398525.

[32] H. Zhang, D.T. Nguyen, H. Zhang, M.T. Thai, Least cost influence maximization across multiple social networks, IEEE/ACM Trans. Netw. 24 (2) (2016) 929–939, http://dx.doi.org/10.1109/TNET.2015.2394793.

[33] H. Sun, X. Gao, G. Chen, J. Gu, Y. Wang, Multiple influence maximization in social networks, in: Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication, in: IMCOM '16, ACM, New York, NY, USA, 2016, pp. 44:1–44:8, http://dx.doi.org/10.1145/2857546.2857591.

[34] L. Guo, D. Zhang, G. Cong, W. Wu, K.L. Tan, Influence maximization in trajectory databases, IEEE Trans. Knowl. Data Eng. 29 (3) (2017) 627–641, http://dx.doi.org/10.1109/TKDE.2016.2621038.

[35] G. Li, S. Chen, J. Feng, K.-l. Tan, W.-s. Li, Efficient location-aware influence maximization, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, in: SIGMOD '14, ACM, New York, NY, USA, 2014, pp. 87–98, http://dx.doi.org/10.1145/2588555.2588561, URL http://doi.acm.org/10.1145/2588555.2588561.

[36] M. Gomez-Rodriguez, L. Song, N. Du, H. Zha, B. Schölkopf, Influence estimation and maximization in continuous-time diffusion networks, ACM Trans. Inf. Syst. 34 (2) (2016) 9:1–9:33, http://dx.doi.org/10.1145/2824253.

[37] M. Gomez Rodriguez, B. Schölkopf, Influence maximization in continuous time diffusion networks, in: Proceedings of the 29th International Conference on Machine Learning, Omnipress, New York, NY, USA, 2012, pp. 313–320.

[38] Y. Li, D. Zhang, K.-L. Tan, Real-time targeted influence maximization for online advertisements, Proc. VLDB Endow. 8 (10) (2015) 1070–1081, http://dx.doi.org/10.14778/2794367.2794376.

[39] S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, J. Tang, Online topic-aware influence maximization, Proc. VLDB Endow. 8 (6) (2015) 666–677, http://dx.doi.org/10.14778/2735703.2735706.

[40] S.S. Singh, A. Kumar, K. Singh, B. Biswas, C2im: Community based context-aware influence maximization in social networks, Physica A 514 (2019) 796–818, http://dx.doi.org/10.1016/j.physa.2018.09.142, URL http://www.sciencedirect.com/science/article/pii/S0378437118312822.

[41] Y. Zhu, D. Li, Z. Zhang, Minimum cost seed set for competitive social influence, in: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2016.7524472.

[42] S.-C. Lin, S.-D. Lin, M.-S. Chen, A learning-based framework to handle multi-round multi-party influence maximization on social networks, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '15, ACM, New York, NY, USA, 2015, pp. 695–704, http://dx.doi.org/10.1145/2783258.2783392.

[43] S.S. Singh, K. Singh, A. Kumar, H.K. Shakya, B. Biswas, A survey on information diffusion models in social networks, in: A.K. Luhach, D. Singh, P.-A. Hsiung, K.B.G. Hawari, P. Lingras, P.K. Singh (Eds.), Advanced Informatics for Computing Research, Springer Singapore, Singapore, 2019, pp. 426–439.

[44] F. Buccafurri, G. Lax, A. Nocera, D. Ursino, Discovering links among social networks, in: P.A. Flach, T. De Bie, N. Cristianini (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 467–482.

[45] T. Suganya, S. Thennammai, R. Velusamy, Unique user identification across multiple social network, Int. J. Res. Mark. 8 (2017) 137–142.

[46] J. Vosecky, D. Hong, V. Y. Shen, User identification across multiple social networks, in: 2009 1st International Conference on Networked Digital Technologies, NDT 2009, 2009, pp. 360–365, http://dx.doi.org/10.1109/NDT.2009.5272173.

[47] F. Battiston, V. Nicosia, V. Latora, Structural measures for multiplex networks, Phys. Rev. E 89 (2014) 032804.

[48] M. Gong, J. Yan, B. Shen, L. Ma, Q. Cai, Influence maximization in social networks based on discrete particle swarm optimization, Inform. Sci. 367–368 (2016) 600–614, http://dx.doi.org/10.1016/j.ins.2016.07.012, URL http://www.sciencedirect.com/science/article/pii/S002002551630500X.

[49] K. Saito, M. Kimura, K. Ohara, H. Motoda, Efficient discovery of influential nodes for sis models in social networks, Knowl. Inf. Syst. 30 (3) (2012) 613–635, http://dx.doi.org/10.1007/s10115-011-0396-2.

[50] Y. Zhang, Influence Maximization on Multi-Phased Multi-Layered Network, 2015.

[51] C. Wang, P. Huang, D. Yang, W. Chen, Cross-layer design of influence maximization in mobile social networks, CoRR abs/1604.02796. arXiv:1604.02796. URL http://arxiv.org/abs/1604.02796, 2016.

[52] F. Erlandsson, P. Bródka, A. Borg, Seed selection for information cascade in multilayer networks, CoRR abs/1710.04391. arXiv:1710.04391. URL http://arxiv.org/abs/1710.04391, 2017.

[53] W.-S. Yang, S.-X. Weng, Application of the ant colony optimization algorithm to the influence-maximization problem, Int. J. Swarm Intell. Evol. Comput. 1 (2012).

[54] M. De Domenico, A. Lima, P. Mougel, M. Musolesi, The anatomy of a scientific rumor, Sci. Rep. 3 (2013) 2980, http://dx.doi.org/10.1038/srep02980.

[55] D.T. Nguyen, H. Zhang, S. Das, M.T. Thai, T.N. Dinh, Least cost influence in multiplex social networks: Model representation and analysis, in: 2013 IEEE 13th International Conference on Data Mining, 2013, pp. 567–576, http://dx.doi.org/10.1109/ICDM.2013.24.

Shashank Sheshar Singh received M.Tech. degree in Computer Science and Engineering from Indian Institute of Technology, Roorkee (IITR). He received B.Tech. degree in Computer Science and Engineering from Kali Charan Nigam Institute of Technology (KCNIT), Banda affiliated to GBTU University, Lucknow. He is working toward the Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi. His research interests include Data Mining, Influence Maximization, Link Prediction and Social Network Analysis.

Kuldeep Singh completed his Ph.D in Computer science & Engineering from Indian Institute of Technology (BHU) Varanasi. His research interest includes High utility itemsets mining, social network analysis, and data mining. He received his M.Tech degree in Computer science and Engineering from Guru Jambheshwar University of Science and Technology, Hisar (Haryana). He has 8 years of teaching and research experience.

Ajay Kumar completed his master of technology in Computer Science & Engineering from Samrat Ashok Technological Institute Vidisha (M.P.) and Bachelor of Technology in Computer Science & Engineering from R.K.D.F Institute of Science and Technology Bhopal (M.P.). He is pursuing Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi. His research interests include Link Prediction and Influence Maximization in social/complex networks.

Bhaskar Biswas received Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi. He received the B.Tech. degree in Computer Science and Engineering from Birla Institute of Technology, Mesra. He is working as Associate Professor at Indian Institute of Technology (BHU), Varanasi in the Computer Science and Engineering department. His research interests include Data Mining, Text Analysis, Machine Learning, Influence Maximization, Link Prediction, Social Network Analysis.