

# BCIM: Budget and capacity constrained influence maximization in multilayer networks

Su-Su Zhang, Chuang Liu, Huijuan Wang, Yang Chen\*, Xiu-Xiu Zhan\*

**Abstract:** Influence maximization (IM) seeks to identify a seed set that maximizes influence within a network, with applications in areas such as viral marketing, disease control, and political campaigns. The budgeted influence maximization (BIM) problem extends IM by incorporating cost constraints for different nodes. However, the current BIM problem, limited by budget alone, often results in the selection of numerous low-cost nodes, which may not be applicable to real-world scenarios. Moreover, considering that users can transmit information across multiple social platforms, solving the BIM problem across these platforms could lead to more optimized resource utilization. To address these challenges, we propose the Budget and Capacity Constrained Influence Maximization (BCIM) problem within multilayer networks and introduce a Multilayer Multi-population Genetic Algorithm (MMGA) to solve it. The MMGA employs modules, such as initialization, repair, and parallel evolution, designed not only to meet budget and capacity constraints but also to significantly enhance algorithmic efficiency. Extensive experiments on both synthetic and empirical multilayer networks demonstrate that MMGA improves spreading performance by at least 10% under the two constraints compared to baselines extended from classical IM problems. The BCIM framework introduces a novel direction in influence maximization, providing an effective and efficient solution to the problem.

**Key words:** Budget and capacity constrained influence maximization; Multilayer network; Information diffusion

## 1 Introduction

How to increase the breadth and speed of information dissemination has garnered increasing attention among

- Su-Su Zhang is with Research Center for Complexity Sciences, Hangzhou Normal University, Hangzhou 311121, China; E-mail: zhangsusu.zss@gmail.com.
- Chuang Liu is with Research Center for Complexity Sciences, Hangzhou Normal University, Hangzhou 311121, China; E-mail: liuchuang\_1985@126.com.
- Huijuan Wang is with Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands; E-mail: H.Wang@tudelft.nl.
- Yang Chen is with Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200433, China; E-mail: chen yang@fudan.edu.cn.
- Xiu-Xiu Zhan is with Research Center for Complexity Sciences, Hangzhou Normal University, Hangzhou 311121, PR China; and also with College of Media and International Culture, Zhejiang University, Hangzhou 310058, PR China; E-mail: zhanxiuxiu@hznu.edu.cn.

\* To whom correspondence should be addressed.

Manuscript received: 2024-08-22; revised: 2024-12-10; accepted: 2025-04-17

researchers due to its wide-ranging applications, such as promoting company products, mitigating negative impacts during crises, and driving social progress and innovation<sup>[1,2]</sup>. The increase in information diffusion breadth is generally referred to as the influence maximization (IM) problem –a classical optimization problem that aims to select a fixed number of nodes to maximize influence spread through a specific diffusion mechanism<sup>[3,4]</sup>. In viral marketing, selecting a group of influential nodes on a social platform can help spread information about a brand or product widely, thus increasing purchase rates<sup>[5]</sup>. Furthermore, seeds derived from solutions to the IM problem under disease spread models can serve as an effective immunization strategy to control contagious diseases, such as COVID-19, H7N9 and AIDS<sup>[6–8]</sup>.

The algorithms for solving the IM problem can be broadly classified into three categories: greedy algorithms, heuristic algorithms, and community-

based algorithms. Greedy algorithms exploit the monotonicity and submodularity of the influence function to achieve an approximately optimal solution, albeit at a high computational cost<sup>[3]</sup>. Over time, numerous studies sought to enhance the efficiency of these algorithms and address specific variants of the IM problem by leveraging marginal gains in innovative ways<sup>[9–12]</sup>. To further improve efficiency, various heuristic methods have been developed<sup>[13,14]</sup>. However, these methods often exhibit strong dependency on the underlying network structure, resulting in considerable performance variability across different networks. Additionally, some researchers have explored community division-based approaches to seed selection, aiming to minimize influence overlap and enhance overall effectiveness<sup>[15,16]</sup>. However, the classical IM problem only seeks optimal seeds to maximize their influence in a network, regardless of the cost of choosing them, resulting in algorithms hardly being applied to real scenarios<sup>[17,18]</sup>. For example, choosing celebrities on Sina Weibo as seeds to promote a product can significantly boost its visibility. However, such choices could lead to a high cost, limiting profits from sales<sup>[19,20]</sup>. Therefore, budgeted influence problem (BIM) has been further studied on single-layer social networks. For instance, Nguyen et al.<sup>[21]</sup> proposed an efficient greedy algorithm which is an approximation of the optimal solution with a certain ratio to the BIM problem, but the algorithm lacks scalability to large networks. Souza et al.<sup>[22]</sup> introduced a new seeding strategy called Node Surround to select influential nodes under the threshold diffusion model, which selects relatively low-cost nodes that are close to expensive and structurally privileged nodes

Even though the above-mentioned methods have successfully solved the BIM problem to some extent in single-layer social networks, there are still some challenges that need to be addressed<sup>[23,24]</sup>. Firstly, previous studies to solve the BIM problem only considered budget constraints, ignoring the number of seeds to be selected. This may result in the selection of a large fraction of nodes as seeds, which is impossible to achieve and also poses a burden in practice. Besides, users may have accounts on different platforms and can transmit information through these social platforms simultaneously<sup>[25–33]</sup>. Consequently, how to select seed nodes to maximize their influence on such multilayer networks needs to be studied.

Therefore, we propose a new and more realistic IM

problem, i.e., budget and capacity constrained influence maximization (BCIM) in multilayer networks, which considers both the budget and the size of the seed set upon the influence maximization problem. To solve the BCIM problem, we propose a multilayer multi-population genetic algorithm (MMGA) that takes into account both the topological properties of a multilayer network and the spreading ability of a node. Specifically, MMGA is inspired by biological evolution and consists mainly of three operators, i.e., crossover, mutation, and selection. In contrast to traditional genetic algorithms, MMGA incorporates a multi-population parallel evolution module to meet seed size constraint and accelerate the efficiency of the algorithm, and a repair module to address situations where the budget is exceeded during the evolution process. Comparative experiments performed on synthetic and empirical multilayer networks under the multilayer independent cascade model show that MMGA surpasses other state-of-the-art benchmarks in terms of influence spread under various constraints. Moreover, we further explore how different parameters, such as crossover probability, mutation probability, evolutionary step, and initialization mode, affect the performance of MMGA. The main contributions of this paper are summarized as follows:

- This study introduces a new and practical problem—the Budget and Capacity Constrained Influence Maximization (BCIM) problem in multilayer networks. This problem uniquely incorporates both budget and seed size as constraints.
- We develop an efficient meta-heuristic algorithm that integrates parallel population evolution and repair modules to effectively solve the BCIM problem.
- A Multilayer Independent Cascade (MIC) model is proposed to simulate the propagation of influence across interactions between entities in different layers, enabling a robust evaluation of the performance of seed sets selected by various algorithms.
- Extensive experiments on real-world and synthetic multilayer networks demonstrate that the proposed algorithm significantly outperforms state-of-the-art baselines in terms of influence spread and robustness.

The remainder of this paper is structured as follows. In Section 2, we provide the preliminary definition of a multilayer network and the spreading dynamics on it. In Section 3, we show the detailed structure of MMGA and use examples to illustrate its key modules. We further illustrate the baselines extended from other IM problems and give detailed descriptions of the empirical multilayer networks that will be used in the subsequent experiment section in Section 4. Furthermore, the effectiveness of the proposed algorithm is evaluated in Section 5. Finally, we conclude this work in Section 6.

## 2 Preliminary definition

### 2.1 Problem Definition

A multilayer network  $G = \{G^1, \dots, G^m, \dots, G^M\}$  contains  $M$  layers of networks, where the  $m$ -th layer is denoted as  $G^m = (V, E^m, C)$ . We assume that each layer contains the same set of nodes, which is denoted as  $V = \{v_1, v_2, \dots, v_N\}$ . Each node may have different connections in different layers, so we use  $E^m = \{e_1^m, e_2^m, \dots, e_h^m\}$  to represent the edge set in the  $m$ -th layer. Multilayer networks are ubiquitous in real-world systems. For instance, on social platforms, users may have various connections representing different types of relationships, such as friendship, kinship, or co-workership<sup>[34]</sup>. By representing each platform as a separate layer, we can construct a multilayer network based on these diverse user connections. Additionally, a multilayer network can represent various types of transportation connections between cities, such as buses, trains, and airlines, with each layer corresponding to a specific mode of transportation<sup>[35]</sup>.

We focus on the budget and capacity constrained influence maximization problem (BCIM), in which the cost of inviting users to do an advertisement or disseminate a piece of information may differ according to the popularity of the users. In real-world scenarios, the cost of activating nodes often varies and is typically correlated with their positions within the network. Research indicates that more influential nodes generally incur higher activation costs. For tasks such as influence maximization and network dismantling, various cost functions have been proposed, many of which are based on a node's degree in the network, as demonstrated in the work of Zhang<sup>[36]</sup> and Ren<sup>[37]</sup>. Accordingly, our study adopts a cost function that is related to node degree. Specifically, we associate each node with a

cost, i.e., we use  $C = \{c_1, c_2, \dots, c_N\}$  to represent the cost of each node and  $c_j = \sum_{m=1}^M d_j^m$ , where  $d_j^m$  is the degree of node  $v_j$  in the network of  $m$ -th layer. This means that we assume that the cost of a node to be chosen as the seed is positively correlated with the sum of the degree of the node in the  $M$  layers. It should be noted that we also address the scenario where node costs are randomly assigned in the experimental section, further demonstrating that our algorithm performs well in both cases: when node costs are correlated with degree and when they are uncorrelated.

The conventional budgeted influence maximization (BIM) problem seeks to maximize influence spread within a budget by selecting seed nodes, but without limiting the number of seeds, which may lead to impractically large sets of low-cost nodes as solutions. For instance, in the context of viral marketing, it would be impractical for marketing entities to rely on numerous low-cost promoters (referred to as seed nodes in the IM problem), as this approach demands considerable time and carries the risk of ineffective promotion. Therefore, we formulate a more practical influence maximization problem, i.e., a budget and capacity constrained influence maximization problem in multilayer networks (BCIM). Given a specific spreading model, the goal of BCIM is to select up to  $K$  nodes whose total cost is with  $B$  as seeds to maximize their influence in a multilayer network, which is mathematically given as follows

$$\sigma(S) = \begin{cases} \operatorname{argmax}_{S \subseteq V} |\bigcup_{m=1}^M V_a^m(S)|, \\ \text{s.t. } |S| \leq K, C(S) \leq B, \end{cases} \quad (1)$$

where  $S$  denotes the seed set, and  $V_a^m(S)$  is a set that contains nodes in  $m$ -th layer that are influenced by the seed nodes in  $S$ ,  $\sigma(S)$  is the influence spread of seed set  $S$ . Moreover,  $C(S) = \sum_{v_i \in S} c_i$  means the cost of seeds and  $B$  is the predefined budget.

### 2.2 Multilayer independent cascade model

To quantify node influence, we utilize the multilayer independent cascade model (MIC) to simulate the spreading process across different layers of a multilayer network. In MIC, nodes can be in one of two states, i.e. active or inactive. If a node is activated in one layer, it will immediately be active in all layers. An inactive node can be activated by an active node with probability  $p$  and remain in the active state permanently. Remarkably, an active node could only activate its

inactive neighbors once. The specific details of MIC are as follows.

- Initially, the nodes in the seed set  $S$  are assigned to the active state, and the remaining nodes are inactive. We use  $S_{t-1}$  to represent the nodes that are newly activated at time step  $t - 1$ , and thus we have  $S_0 = S$ .
- In step  $t$ , for each node  $v_i \in S_{t-1}$ , we first find its inactive neighbors in all layers and name it  $N_{ina}(v_i)$ . We note that  $N_{ina}(v_i)$  may contain repeated nodes  $v_k$ , indicating  $v_k$  being the neighbor of  $v_i$  at different layers. For every node  $v_j \in N_{ina}(v_i)$ , it will be activated by  $v_i$  with probability  $p$ . The newly activated nodes at step  $t$  are inserted into the set  $S_t$ .
- The contagion process continues until there is no new node being activated.

We show an example of the MIC model in Figure 1, where the initial seed node is  $v_3$ . In step  $t = 1$ , the set of inactive neighbors of  $v_3$  in all layers is  $N_{ina}(v_3) = \{v_1, v_2, v_2, v_2, v_4, v_4, v_5, v_5, v_5, v_6, v_7\}$ . Every node in  $N_{ina}(v_3)$  will be activated by  $v_3$  with probability  $p$  and thus  $v_4$  and  $v_5$  are activated and form the newly activated set  $S_1 = \{v_4, v_5\}$ . Subsequently, each node in  $S_1$  will activate their inactive neighbors with probability  $p$  across three layers at time step  $t = 2$ , respectively. At the end of step  $t = 2$ , we get  $S_2 = \{v_2, v_6\}$ . At time step  $t = 3$ , the newly activated nodes of the three layers are  $v_1, v_7$  and  $v_8$ .

**Theorem 1** The BCIM problem under the MIC model is NP-hard.

**Proof.** Consider an instance of the NP-complete Set Cover problem, where we are given a collection of subsets  $S = \{S_1, S_2, \dots, S_{m_1}\}$ , each of which is a subset of a ground set  $U = \{u_1, u_2, \dots, u_{n_1}\}$ .

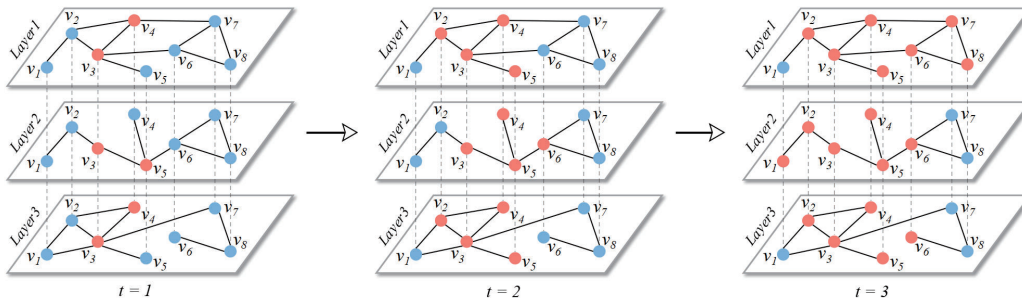
The objective is to determine whether there exist  $K$  subsets in  $S$  whose union covers the entire set  $U$ . We demonstrate that this problem is a special case of our BCIM problem by constructing an equivalent instance as follows.

We construct a three-layer network consisting of  $n_1 + m_1$  nodes. The node set of the first layer is defined as  $V^1 = \{v_1^1, v_2^1, \dots, v_{n_1}^1, v_{n_1+1}^1, v_{n_1+2}^1, \dots, v_{n_1+m_1}^1\}$ , where each node  $v_i^1 (i \in [1, n_1])$  corresponds to an element  $u_i$  of  $U$ , and each node  $v_{n_1+j}^1 (j \in [1, m_1])$  corresponds to a subset  $S_j$  of  $S$ . An edge  $(v_j^1, v_i^1)$  with activation probability  $p = 1$  is added whenever  $u_i \in S_j$ . The second and third layers of the multilayer network are constructed using the same set of nodes  $V^1$  but with different edge configurations in each layer. According to the propagation mechanism of the MIC model, a node activated in the first layer is automatically activated in the other layers. Therefore, the Set Cover problem of the three-layer network is equivalent to identifying a set  $A$  of  $K$  nodes in  $V^1$  such that the influence  $\sigma(A) \geq n_1 + K$  in the first layer.

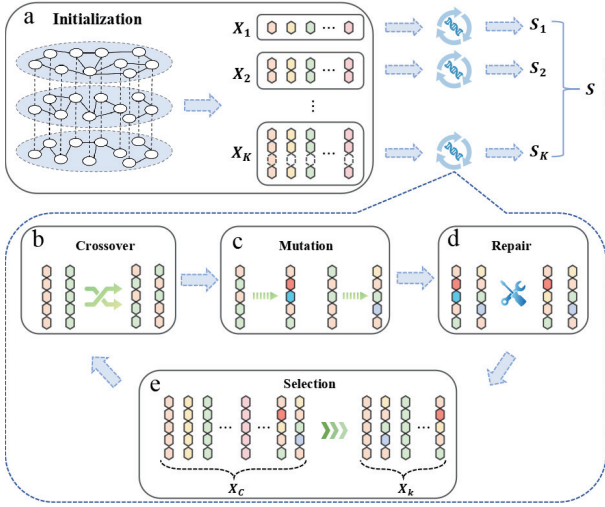
In this context, activation is defined as a deterministic process, and the budget  $B$  is sufficient to select any arbitrary  $K$  nodes from  $V^1$ . If there exists a set of  $K$  nodes in  $V^1$  capable of influencing all nodes corresponding to the ground set  $U$ , those nodes must be part of the set  $S$ . Consequently, if any set  $A$  of  $K$  nodes in  $V^1$  satisfies  $\sigma(A) \geq n_1 + K$ , it implies that the Set Cover problem is solvable. Therefore, the BCIM problem is proven to be NP-hard.

**Theorem 2** For an arbitrary instance of the Multilayer Independent Cascade Model, the resulting influence function  $\sigma(\cdot)$  is submodular.

**Proof.** We denote the marginal gain of any node  $v$  joining the set  $S$  as  $\Delta(S, v) = \sigma(S \cup \{v\} - \sigma(S))$ . Due to the inherent randomness of the propagation process, obtaining a deterministic value for  $\Delta(S, v)$  is



**Fig. 1** A schematic diagram of the MIC model in a three-layered network. We use red and blue to represent nodes in active and inactive states, respectively.



**Fig. 2** Framework of the multilayer multi-population genetic algorithm. (a) The initialization module generates  $K$  populations. Each population contains  $n$  individuals, which are selected from the multilayer network. (b) The crossover module exchanges certain nodes within individuals according to specific rules to generate new offspring. (c) The mutation module alters the nodes within the offspring generated in step (b) to randomly selected nodes. (d) The repair module aims to adjust the offspring that exceed the budget  $B$ . (e) The selection process retains the offspring with the higher fitness value as input for the next iteration.

not feasible. To address this, we employ an equivalent approach to replace the propagation process, enabling us to compute a fixed value for the marginal gain  $\Delta(S, v)$ .

The propagation in the MIC model depends solely on the live state of the edges. In other words, whether the live state of each edge is determined progressively during the propagation process or globally at the beginning does not affect the final propagation size. Therefore, the state of the edges in each layer can be determined by flipping a coin. After this coin-flipping process, we obtain a single-layer live graph that contains all the live edges selected from the multilayer network. We assert that a node  $v$  becomes active if and only if there exists a path from  $S$  to  $v$  consisting entirely of live edges.

Let  $X$  be a sample point in the probability space of coin flip outcomes for all edges. Then,  $\sigma_X(S)$  represents the total number of activated nodes when the seed node set is  $S$  and the coin flip outcomes on the edges are  $X$ . Let  $R(v, X)$  denote the set of nodes that can be reached from  $v$  on a path consisting entirely of live edges. Therefore, the value of  $\sigma_X(S)$  is deterministic and can be explicitly expressed as  $|\cup_{v' \in S}$

$R(v', X)|$ .

Given a set  $S$ , we use  $\sigma_X(S \cup \{v\}) - \sigma_X(S)$  to represent the number of nodes in  $R(v, X)$  that are not included in  $\cup_{v' \in S} R(v', X)$ . Consequently, the inequality  $\sigma_X(S \cup \{v\}) - \sigma_X(S) \geq \sigma_X(T \cup \{v\}) - \sigma_X(T)$  holds for  $S \subseteq T$ , which is the defining inequality for submodularity. Since the influence function  $\sigma$  is the weighted average conditioned on sample  $X$ , i.e.,  $\sigma = \sum_X (\text{Prob}[X] \cdot \sigma_X(S))$ , and  $\sigma_X(S)$  is submodular, it follows that  $\sigma(S)$  is also submodular.

---

**Algorithm 1:** Multilayer Multi-population Genetic Algorithm (MMGA)

---

**Input :**  $G, B, K, p_c, p_m, T$

**Output:**  $S$

```

1  $O \leftarrow \emptyset$ ;
2 for each  $k \leq K$  do
3    $X_k \leftarrow \text{Initialization}(G, V, k, n)$ ;
4   for each  $x \in X_k$  do
5      $x \leftarrow \text{Repair}(x, V, B)$ ;
6   end
7    $x_{k0} \leftarrow$  randomly select from  $X_k$ ;
8   for each  $t \leq T$  do
9      $X_{os} \leftarrow \text{Crossover}(X_k, k, p_c)$ ;
10     $X_C \leftarrow \text{Mutation}(V, X_{os}, X_k, p_m)$ ;
11    for each  $x \in X_C$  do
12       $x \leftarrow \text{Repair}(x, V, B)$ ;
13    end
14     $X_k \leftarrow \emptyset$ ;
15    while  $|X_k| < n$  do
16       $x_i \leftarrow \text{argmax}_{x \in X_C} \{f(x)\}$ ;
17       $X_k \leftarrow X_k \cup x_i$ ;
18      remove  $x_i$  from  $X_C$ ;
19    end
20     $x_{k0} \leftarrow \text{argmax}\{\sigma(x_{ki}), x_{ki} \in \{x_{k0}\} \cup X_k\}$ ;
21  end
22   $O \leftarrow O \cup x_{k0}$ ;
23 end
24  $S \leftarrow \text{argmax}\{\sigma(x), x \in O\}$ 

```

---

### 3 Multilayer Multi-population Genetic Algorithm (MMGA)

The introduction of budget constraints in the BCIM problem can pose challenges for a greedy algorithm—despite its high computational complexity—in finding an optimal solution, particularly when the budget is insufficient. While low-complexity heuristic algorithms might provide feasible solutions, they often do so with a reduced accuracy. To address this issue, we have developed

a meta-heuristic algorithm known as the Multilayer Multi-Population Genetic Algorithm (MMGA), which emulates natural evolutionary mechanisms to search for the global optimal solution. To the best of our knowledge, this is the first time that multi-population evolution has been applied to the BCIM problem, which traditional genetic algorithms cannot effectively solve. Specifically, MMGA comprises initialization, repair and evolution modules, where each population evolves in parallel. We show the framework and pseudo-code of MMGA in Figure 2 and Algorithm 1-5. The details of each module in MMGA are given in the following subsections.

### 3.1 Initialization

In the initial state, we generate a set that contains  $K$  initial populations, i.e.,  $\{X_1, X_2, \dots, X_K\}$ . Each population  $X_k = \{x_{k1}, x_{k2}, \dots, x_{kn}\}$  consists of  $n$  individuals, and every individual is composed of a batch of nodes selected from the multilayer network as the initial solution for the BCIM problem. Specifically, for each individual  $x_{ki} = \{v_{i1}, v_{i2}, \dots, v_{ik}\} \in X_k$ , it contains  $k$  non-duplicate nodes generated from  $V$ . As each  $X_k$  contains  $n$  individuals, we use three different methods, i.e., degree, influence-cost ratio, and random, to initialize the first  $\lceil \frac{n}{3} \rceil$ , the second  $\lceil \frac{n}{3} \rceil$  and the last  $n - 2 \lceil \frac{n}{3} \rceil$  individuals in  $X_k$  in order to accelerate the convergence of the algorithm. The details of the methods are given below, and a more intuitive explanation is given in Figure 3.

**i) Initialization with node degree.** For every individual  $x_{ki} \in X_k$  ( $i \in [1, \lceil \frac{n}{3} \rceil]$ ), we first choose the  $k$  nodes with the highest degree as elements of  $x_{ki}$ . Then each node in  $x_{ki}$  is replaced by a randomly chosen node in  $V \setminus x_{ki}$  with probability  $\beta$ .

**ii) Initialization with influence-cost ratio (ICR).** The influence-cost ratio of nodes  $v_i$  is defined as  $\phi(v_i) = \frac{\sigma(v_i)}{c_i}$ , where  $\sigma(v_i)$  represents the actual spreading capability of node  $v_i$ , quantified by the final outbreak size in the population when selecting node  $v_i$  as the seed, and  $c_i$  denotes the cost of node  $v_i$ . For every individual  $x_{ki} \in X_k$  ( $i \in [\lceil \frac{n}{3} \rceil + 1, 2 \lceil \frac{n}{3} \rceil]$ ), we first choose the  $k$  nodes with highest influence-cost ratio value as the elements of  $x_{ki}$ , and then each element in  $x_{ki}$  is replaced with a node selected from  $V \setminus x_{ki}$  with probability  $\beta$ .

**iii) Initialization randomly.** For every individual  $x_{ki} \in X_k$  ( $i \in [2 \lceil \frac{n}{3} \rceil + 1, n]$ ), the element are selected randomly from  $V$  but should guarantee that there is no

repeated nodes in  $x_{ki}$ .

---

#### Algorithm 2: Initialization

---

**Input :**  $G, V, k, n$ ;  
**Output:**  $X_k, x_{k\alpha}$ ;

```

1  $X_k \leftarrow \emptyset$ ;
2 for  $i$  from 1 to  $\lceil n/3 \rceil$  do
3    $x_i \leftarrow \text{max\_degree}(G, k)$ ;
4   for  $j$  from 1 to  $k$  do
5      $p = \text{random}(0, 1)$ ;
6     if  $p \leq p_\alpha$  then:
7        $v' \leftarrow \text{randomly select from } V$ ;
8        $x_i[j] \leftarrow v'$ ;
9   end
10   $X_k \leftarrow X_k \cup x_i$ ;
11 end
12 for  $i$  from 1 to  $\lceil n/3 \rceil$  do
13    $x_i \leftarrow \text{max\_ICR}(G, k)$ ;
14   for  $j$  from 1 to  $k$  do
15      $p = \text{random}(0, 1)$ ;
16     if  $p \leq p_\alpha$  then:
17        $v' \leftarrow \text{randomly select from } G$ ;
18        $x_i[j] \leftarrow v'$ ;
19   end
20    $X_k \leftarrow X_k \cup x_i$ ;
21 end
22 for  $i$  from 1 to  $n - 2 * \lceil n/3 \rceil$  do
23    $x_i \leftarrow \text{random\_selection}(G, k)$ ;
24    $X_k \leftarrow X_k \cup x_i$ ;
25 end

```

---

### 3.2 Repair Module

Since the populations initialized above may contain individuals that exceed the predefined budget  $B$ , we further design a repair module to adjust the individuals. The main idea of the repair module is to replace the low-influence-cost ratio score nodes with high-influence-cost ratio score nodes, thereby reducing the individual's cost while ensuring the influence spread. Given an individual  $x_i$  that exceeds budgets, i.e.,  $C(x_i) > B$ , we find the node that has the lowest influence-cost ratio score in  $x_i$  and find a node that has the highest influence-cost in  $V \setminus x_i$  to replace it. The process ends at  $C(x_i) \leq B$ . We show an example of repair operation in Figure 4, where the individual  $x_1$  exceeds the budget. The node  $v_{17}$  in  $x_1$  has the lowest influence-cost ratio value  $\phi(v_{17}) = 6$ , so it is replaced by the node  $v_{14}$  with  $\phi(v_{14}) = 42$ . But we still have  $C(x_1) > B$ . So, node  $v_8$  with the lowest influence-cost ratio  $\phi(v_8) = 15$  is further replaced by node  $v_{24}$ . We give the pseudo-code of the repair module in Algorithm 3.

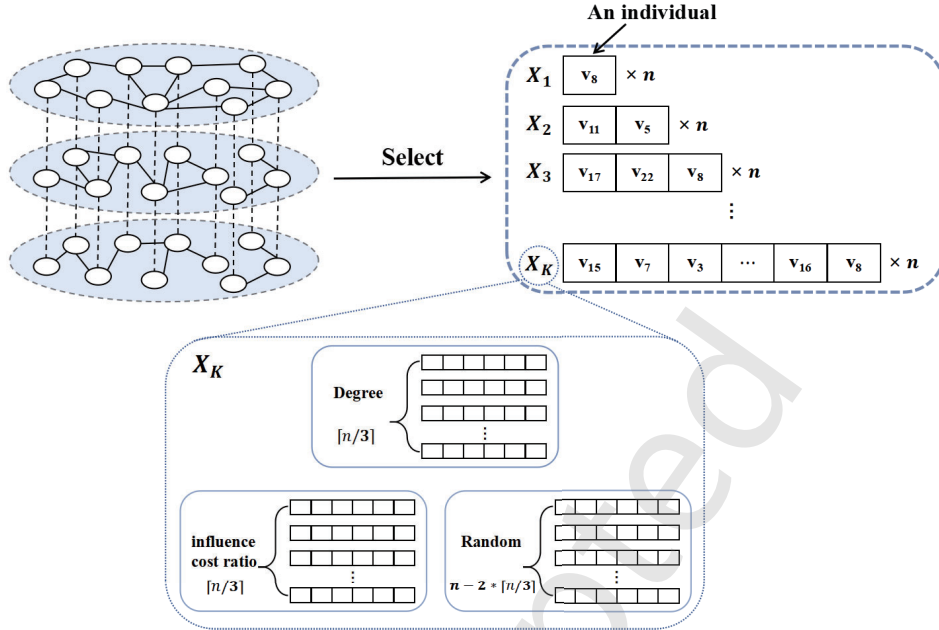


Fig. 3 Initialization of the multilayer multi-population genetic algorithm.

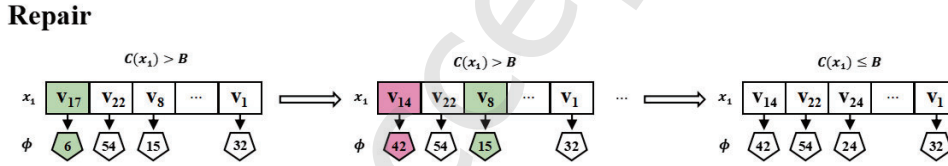


Fig. 4 The repair module in the multi-population genetic algorithm.

**Algorithm 3:** Repair

**Input :**  $x, V, B$ ;  
**Output:**  $x$ ;  
1 **while**  $C(x) > B$  **do**  
2     index  $\leftarrow$  index of node with the lowest ICR in  $x$ ;  
3      $v' \leftarrow$  node with the highest ICR in  $V \setminus x$ ;  
4      $x[index] \leftarrow v'$ ;  
5 **end**

**3.3 Evolution**

To accelerate the computation speed of the algorithm, we apply a parallel evolution strategy, which includes crossover, mutation, and selection, to the initial population set, allowing each population to evolve independently and select its optimal solution. Taking  $X_k(k = 1, 2, \dots, K)$  as an example, we randomly choose an individual  $x_{ko} \in X_k$  as the potential optimal solution in  $X_k$ . Then we update  $x_{ko}$  using the evolution strategy for  $T$  rounds to obtain the final optimal solution for  $X_k$ . Therefore, we can get  $K$  optimal solutions for the  $K$  populations, i.e.,  $O = \{x_{1o}, x_{2o}, \dots, x_{Ko}\}$ ,

using the parallel evolution strategy. Finally, we choose the individual with the highest spread of influence from  $O$  as the final optimal solution for BCIM. Algorithm 4 and 5 show the specific pseudo-code of crossover and mutation. Moreover, we show the details of the evolution strategy using  $X_k(k = 1, 2, \dots, K)$  as an example as follows.

- **Crossover:** The crossover operator generates various combinations of nodes, offering a wide range of potential solutions for the combinatorial optimization problem. Given population  $X_k$ , we randomly choose two individuals from it, namely  $x_{kp} = \{v_{p1}, v_{p2}, \dots, v_{pk}\}$  and  $x_{kq} = \{v_{q1}, v_{q2}, \dots, v_{qk}\}$ . Each corresponding position in  $x_{kp}$  and  $x_{kq}$  is exchanged with probability  $p_c$  and under the condition that no duplicate nodes exist within the resulting offspring individuals, i.e.,  $x'_{kp} = \{v'_{p1}, v'_{p2}, \dots, v'_{pk}\}$  and  $x'_{kq} = \{v'_{q1}, v'_{q2}, \dots, v'_{qk}\}$ . The above process is repeated for  $R$  times, resulting in  $2R$  new offspring.
- **Mutation:** To further expand the solution

**Algorithm 4:** Crossover

---

**Input :**  $X_k, k, p_c$ ;  
**Output:** New offspring  $X_{os}$ ;

```

1  $X_{os} \leftarrow \emptyset$ ;
2 for  $i$  from 1 to  $R$  do
3    $x'_{kp}, x'_{kq} \leftarrow$  randomly select from  $X_k$ ;
4   for  $j$  from 1 to  $k$  do
5      $p = \text{random}(0, 1)$ ;
6     if  $p \leq p_c$  and  $x'_{kp}[j] \notin x'_{kq}$  and  $x'_{kq}[j] \notin x'_{kp}$ 
7       then:
8         switch  $x'_{kp}[j]$  and  $x'_{kq}[j]$ ;
9    $X_{os} \leftarrow X_{os} \cup x'_{kp}$ ;
10   $X_{os} \leftarrow X_{os} \cup x'_{kq}$ ;
11 end

```

---

space, the MMGA algorithm performs a mutation operator on the offspring individuals generated by the crossover operator. In particular, each node in the offspring  $x_i$  is replaced by a random node chosen from  $V \setminus x_i$  with a probability of  $p_m$ . It should be noted that the  $2R$  new offspring generated by crossover and mutation operators may exceed the budget. Hence, we further use the repair module illustrated in the initialization step to further repair the offspring. Finally, a new population  $X_C$  is generated, which contains  $n + 2R$  individuals composed of individuals of the offspring  $X_k$  and  $2R$  generated by the crossover and mutation operator.

For the sake of clarity, we give detailed examples to further visualize the crossover and mutation operators, which are shown in Figure 5. We use a crossover operator to exchange every position in  $x_1$  and  $x_2$ . In the first position, we generate a random number  $y$ , and since  $y$  is smaller than  $p_c$ ,  $v_8$  and  $v_{12}$  are exchanged. However, a random number  $y$  is generated for the node pairs  $v_6$  and  $v_5$  at the second position, they are not swapped since  $y \geq p_c$ . Subsequently, even though  $y$  is smaller than  $p_c$  in the third position for the pair of nodes  $v_{11}$  and  $v_6$ , they cannot exchange as  $v_6$  is in  $x_1$  in the second position. The crossover operator will work on all the positions in  $x_1$  and  $x_2$ , and finally we can obtain two offspring  $x'_1 = \{v_{12}, v_6, v_{11}, \dots, v_3\}$  and  $x'_2 = \{v_8, v_5, v_6, \dots, v_2\}$ . We use offspring  $x'_1$  to illustrate the mutation process, in which each node is mutated with probability  $p_m$ . We generate a random number  $y$  for each node in  $x'_1$

and only mutate the nodes when  $y < p_m$ . For example, node  $v_6$  is replaced by  $v_{16}$ , which is chosen randomly from  $V \setminus x'_1$ .

**Algorithm 5:** Mutation

---

**Input :**  $V, X_{os}, X_k, p_m$ ;  
**Output:**  $X_C$ ;

```

1  $X_C \leftarrow X_k$ ;
2 for  $i$  from 1 to  $2 * R$  do
3   for  $j$  from 1 to  $k$  do
4      $p = \text{random}(0, 1)$ ;
5     if  $p \leq p_m$  then:
6        $v' \leftarrow$  randomly select from  $V \setminus X_{os}[i]$ ;
7        $X_{os}[i][j] \leftarrow v'$ ;
8   end
9    $X_C \leftarrow X_C \cup X_{os}[i]$ ;
10 end

```

---

- **Selection:** The selection module aims to select  $n$  individuals with high fitness scores from  $X_C$  as the new population  $X_k$  for the next iteration. For an individual  $x_i$ , the fitness score is defined as  $f(x_i) = \sigma(x_i)$ , where  $\sigma(x_i)$  represents the final influence spread of  $x_i$  when it is chosen as the seed set under the MIC model. We use the roulette wheel algorithm to select  $n$  individuals from  $X_C$  to form a new population  $X_k$ . Particularly, the probability that  $x_i$  is selected is  $f(x_i) / \sum_{x_j \in X_C} f(x_j)$ , meaning that individuals with high fitness scores have a high chance to be chosen. After we get the updated  $X_k$ , we find the optimal individual in it, i.e.,  $x'_{ko} = \text{argmax}\{\sigma(x_{ki}), x_{ki} \in X_k\}$ . Then, we update  $x_{ko}$  by

$$x_{ko} = \begin{cases} x'_{ko}, & \text{if } x'_{ko} > x_{ko} \\ x_{ko}, & \text{otherwise} \end{cases} \quad (2)$$

In summary, MMGA algorithm initializes multiple populations with prior knowledge through its initialization module to promote rapid convergence. Each population then independently seeks a local optimal solution in parallel within the evolution module. We have also designed a repair function to handle cases where individuals might exceed the budget during the evolution process. Finally, the global optimal solution is selected from the local optimal solutions of each population, forming the final set of seeds.

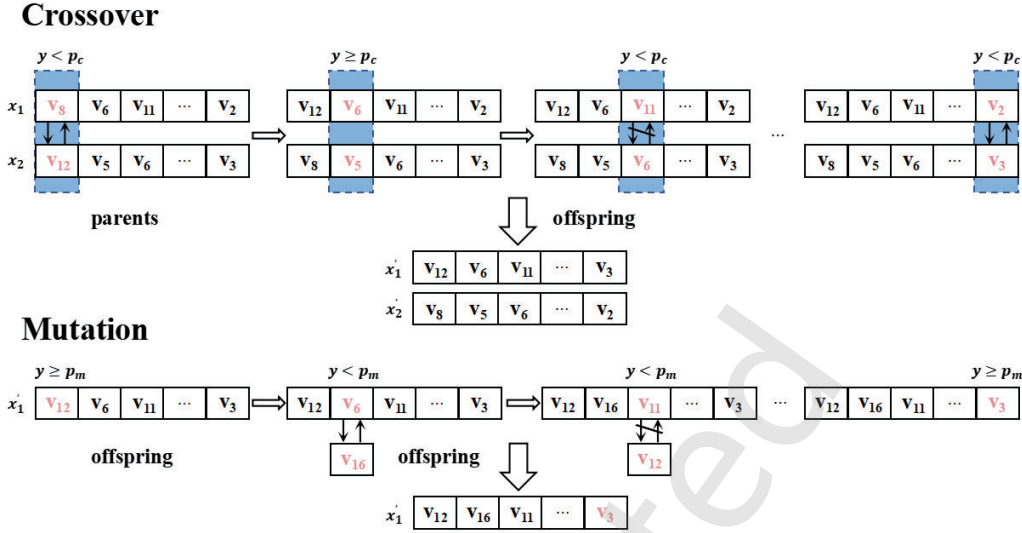


Fig. 5 The evolution of multi-population genetic algorithm.

## 4 Baselines and Datasets

### 4.1 Baselines

To validate the effectiveness of MMGA in solving the BCIM problem, we extend a series of baseline methods, which were originally designed for the conventional influence maximization problem or budgeted influence maximization problem, to solve the BCIM problem. We choose state-of-the-art algorithms, including meta-heuristic algorithms, such as DPSO, Greedy algorithm, and heuristic algorithms, i.e., ACD, Degree, and Community-based solution approach as baselines. DPSO is a representative optimization algorithm frequently used in combinatorial optimization problems. The greedy algorithm has been rigorously proven to achieve an approximation ratio of  $1-1/e$  for the conventional IM problem, making it a challenging benchmark for researchers to surpass. Additionally, ACD is an IM algorithm tailored for multilayer network structures, while Combim is specifically designed for the budgeted influence maximization problem. The details of each algorithm are given below.

#### Discrete Particle Swarm Optimization (DPSO)<sup>[38]</sup>:

We generalize the DPSO algorithm, which is inspired by the self-organizing behavior of bird flocks, from single-layer network to multilayer network as a baseline method. To solve the BCIM problem, the algorithm generates  $n$  particles as  $n$  possible solutions. Here, we set  $n$  to 60, matching the initial population size used in MMGA. The  $n$  particles evolve simultaneously

and the one with the highest fitness score, which is quantified by the final influence spread of the seeds by Monte Carlo simulations, will be chosen as the optimal seed set. Each particle  $i$  is associated with a position vector  $\mathbf{x}_i = (v_{i1}, v_{i2}, \dots, v_{iK})$  which is initialized by degree, with non-duplicate nodes, and a velocity vector  $\mathbf{V}_i = (r_{i1}, r_{i2}, \dots, r_{iK}) (r_{ik} \in \{0, 1\})$  which is initialized by all zero values.

The velocity vector  $\mathbf{V}_i$  determines the optimization direction of the particle  $i$  and is updated by the best personal position vector  $\mathbf{x}_i^{Pbest} = (v_{i1}^{Pbest}, v_{i2}^{Pbest}, \dots, v_{iK}^{Pbest})$  and the best global position vector  $\mathbf{x}^{Gbest} = (v_1^{Gbest}, v_2^{Gbest}, \dots, v_K^{Gbest})$ . At time step  $t$ , for particle  $i$ , if its current fitness score is larger than the historical optimum, the current particle's position vector becomes its best personal position  $\mathbf{x}_i^{Pbest}$ . The position vector of a particle with the highest fitness score in the swarm, i.e., among all particles, is considered the best global solution  $\mathbf{x}^{Gbest}$  at the time step  $t$ . The velocity vector of the particle  $i$  is updated via the following formula

$$\mathbf{V}_i' = \mathbf{H}(w\mathbf{V}_i + I_1 J_1 (\mathbf{x}_i \cap \mathbf{x}_i^{Pbest}) + I_2 J_2 (\mathbf{x}_i \cap \mathbf{x}^{Gbest})), \quad (3)$$

where  $w$  represents the inertia weight, parameters  $I_1$  and  $I_2$  are learn factors,  $J_1$  and  $J_2 \in [0, 1]$  denote random numbers. The operator  $\cap$  is a similar intersection operation, in which if  $v_{ik}$  of  $\mathbf{x}_i$  is also in  $\mathbf{x}_i^{Pbest}$ , then the  $k$ -th position in  $\mathbf{x}_i \cap \mathbf{x}_i^{Pbest}$  is set to 0; otherwise, it is set as 1. In addition,  $\mathbf{H}(\cdot)$  is defined as a threshold function shown as follows

$$\mathbf{H}(q) \begin{cases} 0, & \text{if } q < 2 \\ 1, & \text{if } q \geq 2 \end{cases} \quad (4)$$

Therefore, we can further update particle  $i$  via the updated velocity vector  $\mathbf{V}'_i$  using the following formula

$$\mathbf{x}_i \oplus \mathbf{V}'_i = \mathbf{x}'_i = \{v'_{i1}, v'_{i2}, \dots, v'_{iK}\} \quad (5)$$

$$v'_{ik} = \begin{cases} v_{ik}, & \text{if } r'_{ik} = 0 \\ \text{replace}(v_{ik}, V \setminus x_i), & \text{if } r'_{ik} = 1 \end{cases} \quad (6)$$

The operator  $\oplus$  drives the particles to the promising regions and the function of *replace* means selecting a random node from  $V \setminus x_i$  to replace  $v_{ik}$ . The best global solution for the BCIM problem is obtained after  $T$  iterations of the above procedures.

**Greedy:** We propose a greedy algorithm to solve the BCIM problem, where the influence spread of a node is quantified using Monte Carlo simulations based on the MIC model. The marginal gain of any node  $v$  joining  $S$  is denoted as  $\Delta(S, v) = \sigma(S \cup \{v\}) - \sigma(S)$ . Initially, the node  $v_i$  ( $c_i \leq B$ ) with the highest influence spread  $\sigma(v_i)$ , as evaluated by the MIC spreading model, is selected as the first seed and added to  $S$ . The budget is then updated as  $B = B - c_i$ . Next, the second seed node  $v_j$  is chosen from  $V \setminus \{v_i\}$ , ensuring it achieves the highest marginal gain  $\Delta(S, v_j)$  while satisfying  $c_j \leq B$ . This process is repeated until the seed set reaches the predefined size  $K$  or the budget is exhausted.

**CELF++**<sup>[9]</sup> is an optimized greedy algorithm for the IM problem. In the context of BCIM, the algorithm initializes an empty seed set  $S$ , a heap  $Q$ , and two variables: *last\_seed* and *cur\_best*. The variable *last\_seed* tracks the ID of the last seed selected by the algorithm, while *cur\_best* identifies the node with the maximum marginal gain. Each node  $v$  is associated with a tuple  $\langle v.mg1, v.prev\_best, v.mg2, v.flag \rangle$ , where  $v.mg1 = \Delta_v(S)$  represents  $v$ 's marginal gain when added to  $S$ . The variable *v.prev\_best* refers to the node with the maximum marginal gain before considering  $v$ ,  $v.mg2 = \Delta_v(S \cup \{pre\_best\})$ , and *v.flag* indicates the iteration number.

Initially, *last\_seed* and *cur\_best* are set to null. For each node  $v$ ,  $v.mg1 = \sigma(v)$  is calculated and added to the heap  $Q$  if its cost  $c(v) \leq B$ . Additionally,  $v.mg2 = \sigma(v, cur\_best)$  and  $v.flag = 0$  are recorded. During the iterations, the top element  $v$  is extracted from  $Q$ ,

and its marginal gain is updated based on its *flag* and *last\_seed* as follows:

- If  $v.flag = |S|$ ,  $v.mg1$  is already updated, so  $v$  is directly added to  $S$ , and *last\_seed* is updated to  $v$ .
- If  $v.prev\_best = last\_seed$ ,  $v.mg2$  is assigned to  $v.mg1$ .
- Otherwise,  $v.mg1$  is recalculated and  $v$ 's position in  $Q$  is updated. Moreover,  $v.prev\_best$  and  $v.mg2$  are updated to  $cur\_best$  and  $\Delta_v(S \cup \{cur\_best\})$ , respectively.

For each node  $v$  traversed,  $v.flag$  is set to  $|S|$ , and  $cur\_best$  is updated based on  $v.mg1$ . If node  $v$  is selected as a seed, the remaining budget  $B$  is updated as  $B = B - c(v)$ . The heap  $Q$  is then adjusted to ensure that the cost of all elements in  $Q$  remains below  $B$ . The seed selection process continues until  $S$  reaches the desired size  $K$  or the budget is fully utilized.

**Adaptive Coupling Degree (ACD)**<sup>[39]</sup> algorithm considers the adaptive coupling degree to find seed nodes for BCIM problem. The number of non-repeated neighbors across all the layers of a node  $v_i$  is assigned as the initial value of the adaptive degree of  $v_i$ . In each iteration, ACD selects the node with the highest adaptive coupling degree as the seed node and updates the adaptive degree of each neighbor  $v_q$  of the corresponding seed node using the following formula

$$D'_c(v_q) = D_c(v_q) - e^{Q(v_q)}, \quad (7)$$

where  $Q(v_q)$  represents the number of seed nodes in the neighborhood of node  $v_q$ . The iterative selection process continues until the number of seed nodes reaches  $K$  or the budget is exhausted.

**Degree** assumes that the nodes with more neighbors are more influential in a multilayer network. For node  $v_i$ , we use the sum of the node degree across different layers to quantify its spreading ability, and then the nodes are sorted in descending order based on their spreading ability. Subsequently, we select the nodes with top spreading ability as the seeds until the size of the seed set reaches  $K$  or the budget is exhausted.

**Community-based solution approach for the BCIM problem (ComBim)**<sup>[40]</sup> addresses the BCIM problem based on community structure, which uses the community detection algorithm proposed by Pan et al.<sup>[41]</sup> to detect communities in all layers of a multilayer network. In the community detection algorithm, the

similarity between two nodes  $v_i$  and  $v_j$  is defined as  $Sim_{ij}^m = \sum_{v_z \in N^m(v_i) \cap N^m(v_j)} \frac{1}{d_z^m}$ , where  $N^m(v_z)$  and  $d_z^m$  represent the neighboring set and the degree of node  $v_z$  in the  $m$ -th layer, respectively. After obtaining all communities in the multilayer network, we allocate budgets to these communities according to their size, i.e., communities with larger sizes can get more budgets and vice versa. The selection of seed nodes starts from the smallest community until the  $K$  seed nodes are selected or the budget is exceeded. In each of the communities, we select nodes with high degree as the seeds until the budget of this community is exceeded. The remaining budget of a community is transferred to the community with the most connections to it. The selection process continues until the number of seed nodes reaches  $K$  or the total budget is exhausted.

## 4.2 Empirical and synthetic multilayer networks

We summarize three real-world multilayer networks and three artificial multilayer networks to evaluate the effectiveness of the algorithms discussed in the subsequent sections. The basic topological characteristics of these multilayer networks are presented in Table 1. The empirical multilayer networks are derived from transportation systems, social networks, and biological systems, respectively, and have been widely utilized in studies of resilience analysis, spreading dynamics, and structural reducibility, among other applications<sup>[42–44]</sup>.

### 4.2.1 Empirical multilayer networks

**London Multiplex Transport Network (Transport).**<sup>[45]</sup> The data was collected in 2013, which uses edges to represent whether there is a route between two stations (i.e., nodes in the network) or not. It contains three layers, illustrating stations connected by underground line (U), overground (O) and Docklands light railway (D), respectively.

**CKM Physicians Innovation Network (CKM).**<sup>[46]</sup> The edges in this multilayer network were collected among physicians in four towns in Illinois, i.e., Peoria, Bloomington, Quincy and Galesburg. Each layer is constructed based on each of the following questions:

- i) Who do you typically consult when you need advice about therapy?
- ii) Who are the three or four physicians with whom you generally discuss cases or treatments during a typical week, such as last week?
- iii) Could you provide the first names of the three or four people you spend the most social time with?

The three layers are named shortly Q.1, Q.2, and Q.3 in the following study, respectively.

**Drosophila Multiplex Gpi Network (Drosophila).**<sup>[47]</sup> In this data, we use edges to represent the interactions between genetics and proteins. The interactions in the three layers are direct interactions (DI), suppressive interaction (SI), and Physical association (Phy), respectively.

### 4.2.2 Synthetic multilayer networks

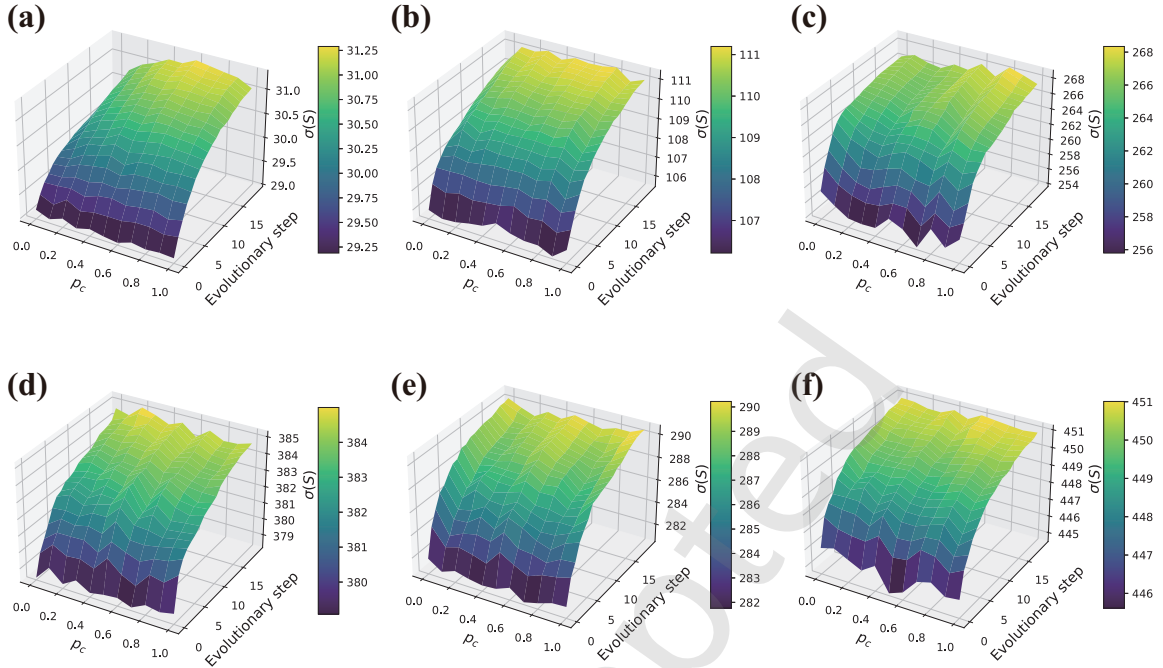
**ER Multilayer Network.** The construction of a single-layer ER network starts with a graph with  $N = 1000$  isolated nodes. For each pair of nodes  $(v_i, v_j)$  where  $v_i \neq v_j$ , we add an edge between them with a probability of 0.004. We generate three ER networks with the same set of nodes to form a three-layer ER network.

**WS Multilayer Network.** The single-layer WS network starts with a ring structure of 1000 nodes, with each node connecting to its four nearest neighbors. To introduce randomness, we rewire each edge with a probability of 0.3. This process is repeated to create three WS networks with the same set of nodes and finally form a three-layer WS network.

**BA Multilayer Network.** To construct the BA network, we start with three initial nodes connected by two edges. At each time step, a new node is introduced and connects to two existing nodes using a preferential attachment mechanism, i.e., connecting to nodes with high degree. The process continues until the network reaches 1000 nodes. We use the same node set to create three BA networks to form a three-layer BA network.

## 5 Experiment

To validate the effectiveness of the MMGA algorithm, we conduct experiments on the three empirical multilayer networks and three types of synthetic multilayer networks. All experiments are carried out under the MIC model with a propagation probability  $p = 0.1$  unless otherwise specified, and the influence spread of a seed set is quantified by the average of the final outbreak size of 1000 Monte Carlo simulations. To assess the effectiveness of the proposed method, we change the value of the seed size  $K$  from 10 to 20 with an interval of 2, and the value of the budget  $B$  from 200 to 600 with an interval of 50. All the algorithms are implemented in Python and run independently on a server with a 2.20GHz Intel(R) Xeon(R) Silver 4114 CPU and 90GB of memory.



**Fig. 6** Performance of MMGA with the change of crossover probability  $p_c$  and evolutionary step. We set  $p_m = 0.005$ ,  $K = 20$ , and  $B = 600$ . The results are given for networks: (a) Transport; (b) CKM; (c) Drosophila; (d) ER; (e) WS; and (f) BA.

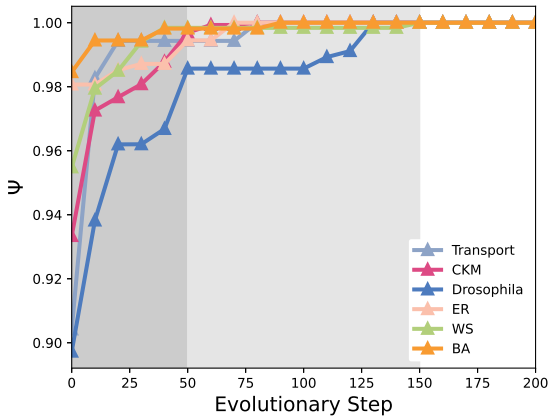
**Table 1** Topological characteristics of the empirical multilayer networks, in which we use  $N$  and  $|E|$  to represent the number of nodes and edges of each layer,  $\langle d \rangle$  denotes the average degree of nodes,  $\langle l \rangle$  is the average shortest path,  $L$  represents the diameter,  $C$  represents the clustering coefficient and  $D$  represents the link density of each layer.

Dataset	Layer	$N$	$ E $	$\langle d \rangle$	$\langle l \rangle$	$L$	$C$	$D$
Transport	U	271	312	2.3	13.96	39	0.031	0.0085
	O	83	83	2.0	13.49	35	0.0	0.0244
	D	45	46	2.04	8.23	23	0.019	0.0465
CKM	Q.1	215	449	4.18	2.90	7	0.260	0.0195
	Q.2	231	498	4.31	2.94	8	0.260	0.0187
	Q.3	228	423	3.71	3.45	9	0.211	0.0163
Drosophila	DI	3126	5472	3.50	0.99	13	0.024	0.0011
	SI	239	270	2.26	1.30	7	0.174	0.0095
	Phy	120	160	2.67	1.41	9	0.016	0.0224
ER	ER.1	1000	1974	3.95	0.38	11	0.004	0.0040
	ER.2	1000	1996	3.99	0.37	11	0.005	0.0040
	ER.3	1000	1994	3.99	0.34	11	0.002	0.0040
WS	WS.1	1000	2000	4.0	6.13	11	0.179	0.0040
	WS.2	1000	2000	4.0	6.19	11	0.180	0.0040
	WS.3	1000	2000	4.0	6.23	11	0.199	0.0040
BA	BA.1	1000	1996	3.99	4.03	7	0.026	0.0040
	BA.2	1000	1996	3.99	3.96	7	0.042	0.0040
	BA.3	1000	1996	3.99	4.13	8	0.017	0.0040

## 5.1 Parametric analysis

In MMGA, parameters such as the probability of crossover  $p_c$ , the evolutionary step, and the probability of mutation  $p_m$  would affect the influence spread of seeds. Therefore, we first explore the impact of different crossover probabilities  $p_c$  on the influence spread of seeds when the evolutionary time step changes in both empirical and synthetic multilayer networks. The remaining parameters are set as follows: mutation probability  $p_m = 0.005$ , budget  $B = 600$ , and seed set  $K = 20$ . As shown in Figure 6, the spread of influence fluctuates with an increase of  $p_c$  for different values of evolutionary steps and networks. The high value of the spread of the influence is achieved when  $p_c$  is in the range of 0.6 to 0.9. We also show how the influence spread changes with the change of the evolutionary step in Figure 6 and Figure 7. For clarity, we use the normalized influence spread (i.e., normalized by network size) as the y-axis in Figure 7, and the probabilities of crossover and mutation are set to 0.8 and 0.005, the constraints of budget and seed set size are set to 600 and 20, respectively. As shown in the two figures, the spread of influence increases dramatically when the evolutionary step increases. More concretely, when the evolutionary step is less than 50, the performance of MMGA improves rapidly. In most multilayer networks, the algorithm performance

enters a gradual improvement phase between 50 and 150 steps, and after the 150 steps it becomes stable. Consequently, we use the probability of crossover  $p_c$  and the evolutionary step as 0.8 and 150 in subsequent experiments, respectively.

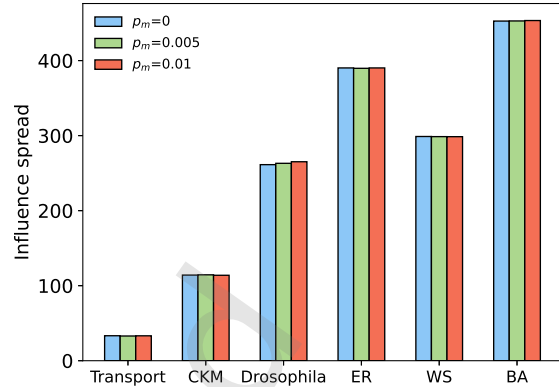


**Fig. 7** The normalized influence spread  $\Psi$  of MMGA when the evolutionary step changes. We set  $p_c = 0.8$ ,  $p_m = 0.005$ ,  $K = 20$ , and  $B = 600$ . The dark gray and light gray areas represent the rapid and gradual improvement phases of algorithm performance, respectively.

In previous studies, the probability of mutation  $p_m$  was generally low. We test the performance of the algorithm setting  $p_m$  as 0, 0.005, and 0.01, where the results are given in Figure 8. We observe that the performance of MMGA with  $p_m = 0.005$  and 0.01 is slightly better than that of  $p_m = 0$ . And the performance of  $p_m = 0.005$  and 0.01 is similar in different networks. Therefore, we choose  $p_m = 0.005$  in the following experiment.

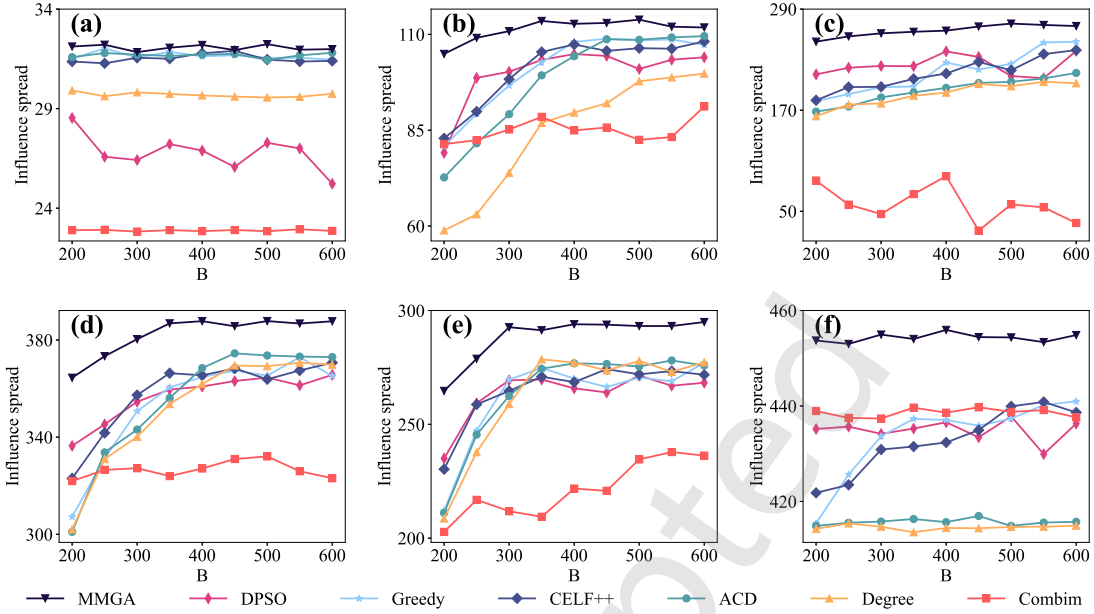
## 5.2 Algorithm performance

The performance of the algorithms under different budgets and seed sizes is shown in Figure 9 and Figure 10, respectively. In Figure 9, we set the seed size as  $K = 20$ , and we set the budget as  $B = 600$  in Figure 10. Figure 9 shows that MMGA could always find the optimal seed set to maximize the spread of influence with different budget values  $B$ . It performs better than the baseline methods, especially for small values of  $B$ . DPSO performs second best on a small budget in most networks. However, MMGA outperforms DPSO in all networks by almost 10% when  $B \leq 300$ . Greedy and CELF++ exhibit comparable performance, performing relatively well on the Transport network but demonstrating



**Fig. 8** The average influence spread of the seeds selected by MMGA under different mutation probability  $p_m$ . We set  $p_c = 0.8$ ,  $K = 20$ ,  $B = 600$ , and  $T = 500$ .

unstable performance on other networks. Without budget constraints, greedy-based algorithms naturally prioritize high-influence nodes. However, in the context of BCIM, high-influence nodes are typically associated with higher costs. As a result, when the budget is limited, selecting these nodes may exceed the budget, forcing the greedy-based algorithms to choose nodes with relatively lower influence spread but within the budget constraints. This trade-off often leads to suboptimal performance compared to MMGA. This trend is further illustrated in Figure 9, which shows that the performance of both the greedy algorithm and CELF++ improves as the budget increases. For degree-based benchmarks, i.e., ACD and Degree, MMGA achieves an average improvement of 12.8% and 15%, respectively. The selection of seeds of Combim starts from the community with the smallest size, and once the number of nodes reaches  $K$ , it can no longer choose additional nodes. In other words, the budget is not fully utilized, resulting in the algorithm's poor performance. Overall speaking, the spread of influence in different networks is relatively stable for MMGA, whereas most of the baselines show an increasing trend in most networks with the increase in budget. This implies that the baseline methods are more sensitive to budget changes compared to MMGA. Similar to Figure 9, MMGA also performs best when we change the seed set size constraint  $K$  in different networks, as shown in Figure 10. With an increase in  $K$ , the spread of influence increases for most algorithms. The performance of the baselines is consistent with those in Figure 9.



**Fig. 9** The influence spread of different algorithms under various budget conditions, in which we set  $K = 20$ . The results are given for networks: (a) Transport; (b) CKM; (c) Drosophila; (d) ER; (e) WS; and (f) BA.

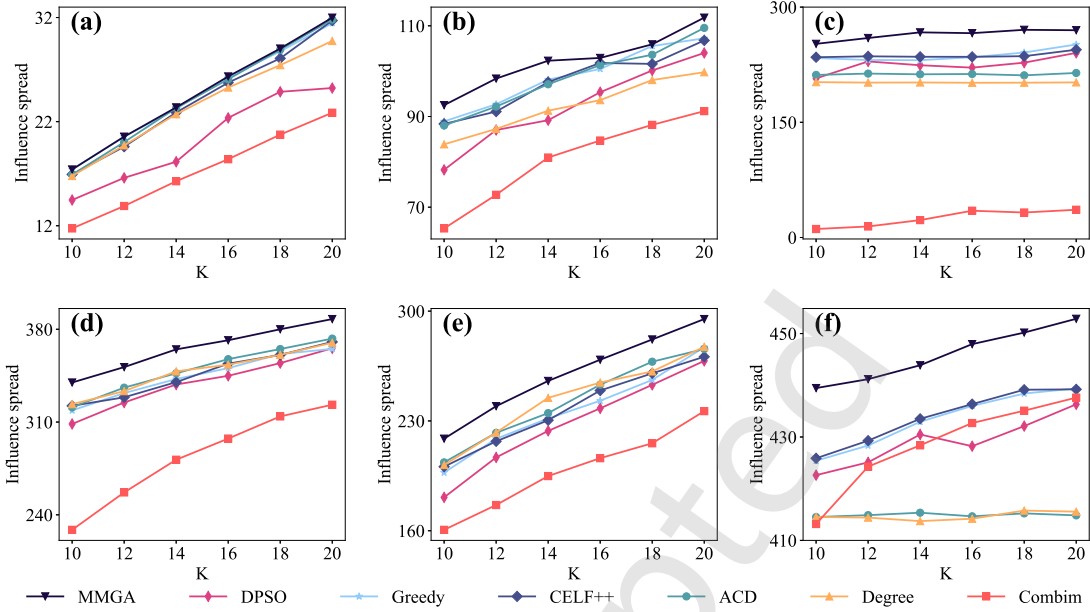
To further explore the robustness of MMGA, we increase the probability of propagation of the MIC model to larger values, such as  $p = 0.3$  and  $0.5$ , and keep the budget and seed size as  $B = 600$  and  $K = 20$ , respectively. The spread of influence of the seed set obtained by different algorithms is given in Table 2, where the best performance is shown in bold, and the second best is indicated with an underline. Generally speaking, when the propagation probability is high, the strong spreading coupling of multilayer networks narrows the performance differences between different algorithms, which is particularly evident in dense synthetic networks, such as WS and BA networks. Furthermore, the table shows that MMGA outperforms the other benchmarks under the MIC model regardless of how  $p$  is set. The CELF++ algorithm exhibits slightly better performance compared to the classic Greedy algorithm. Meanwhile, Combin achieves the second-best result, with ACD following closely behind. Both the Combin and ACD algorithms fundamentally adopt the concept of reducing influence overlap, their advantages becoming more apparent as the propagation probability increases.

Finally, we consider a scenario where node costs are decoupled from node degrees, i.e., node costs are randomly assigned. For a given multilayer network, we randomly shuffle the original node costs (initially derived from degrees), ensuring that

the overall cost distribution remains unchanged while making the node costs independent of node degrees. Figure 11 illustrates the performance of each algorithm under this randomized cost assignment, with results averaged over 50 reshufflings of the node costs. The results show that MMGA consistently outperforms all baseline algorithms across different multilayer networks. Moreover, under the decoupled cost assignment, greedy-based algorithms exhibit the second-best performance after MMGA (except in Figure 11(c)), likely because greedy-based algorithms are more inclined to prioritize highly influential yet low-cost nodes.

### 5.3 Ablation experiment

In the initialization of the population in MMGA, we use a combination of degree, influence-cost ratio (ICR), and random strategy to generate the populations. We investigate the influence of this combination on the final performance of MMGA, in which we show the AUC values of different initialization methods, i.e., Mixed (combination of degree, influence-cost ratio (ICR), and random strategy), Degree, ICR, and Random, when keeping  $K$  unchanged. The AUC value is the area under the influence spread curve when the value of the budget  $B$  changes. As shown in the table, the method that combines Degree, SRC, and Random initialization consistently achieves the best results, highlighted in



**Fig. 10** The influence spread of different algorithms under various seed set sizes, in which we set  $B = 600$ . The results are given for networks: (a) Transport; (b) CKM; (c) Drosophila; (d) ER; (e) WS; and (f) BA.

**Table 2** The influence spread of different algorithms under various propagation probability and we set  $K = 20$  and  $B = 600$ . The best performance is shown in bold and the second best is shown with an underline in each multilayer network.

	p=0.3						p=0.5					
	Transport	CKM	Drosophila	ER	WS	BA	Transport	CKM	Drosophila	ER	WS	BA
MMGA	<b>68.91</b>	<b>228.57</b>	<b>1617.63</b>	<b>965.75</b>	<b>981.12</b>	<b>947.35</b>	<b>125.37</b>	<b>239.36</b>	<b>2390.26</b>	<b>996.01</b>	<b>999.56</b>	<b>995.45</b>
DPSO	54.09	221.30	<u>1604.02</u>	957.27	980.73	945.01	114.36	235.84	2372.94	992.03	999.51	995.24
Greedy	61.67	221.51	1594.19	957.62	980.85	946.02	113.92	235.82	2363.28	991.80	<u>999.56</u>	995.30
CELF++	61.72	<u>224.55</u>	1590.52	958.08	980.90	946.01	113.57	<u>238.84</u>	2359.77	<u>995.08</u>	999.51	995.43
ACD	<u>67.39</u>	221.06	1587.89	957.13	980.20	944.46	<u>119.02</u>	236.01	2359.59	991.79	<u>999.56</u>	995.26
Degree	58.67	221.08	1585.15	957.61	980.81	943.85	104.91	235.94	2361.98	991.95	999.47	995.16
Combim	37.72	<u>222.35</u>	1596.63	<u>959.59</u>	<u>980.94</u>	<u>946.32</u>	79.64	236.38	<u>2374.88</u>	992.79	999.46	<u>995.45</u>

bold, in different settings of seed set sizes. In most cases, the random strategy achieves the second-best performance. Actually, using the combination of the three different strategies cannot only achieve better performance in influence spread, but also could help accelerate the convergence speed of MMGA.

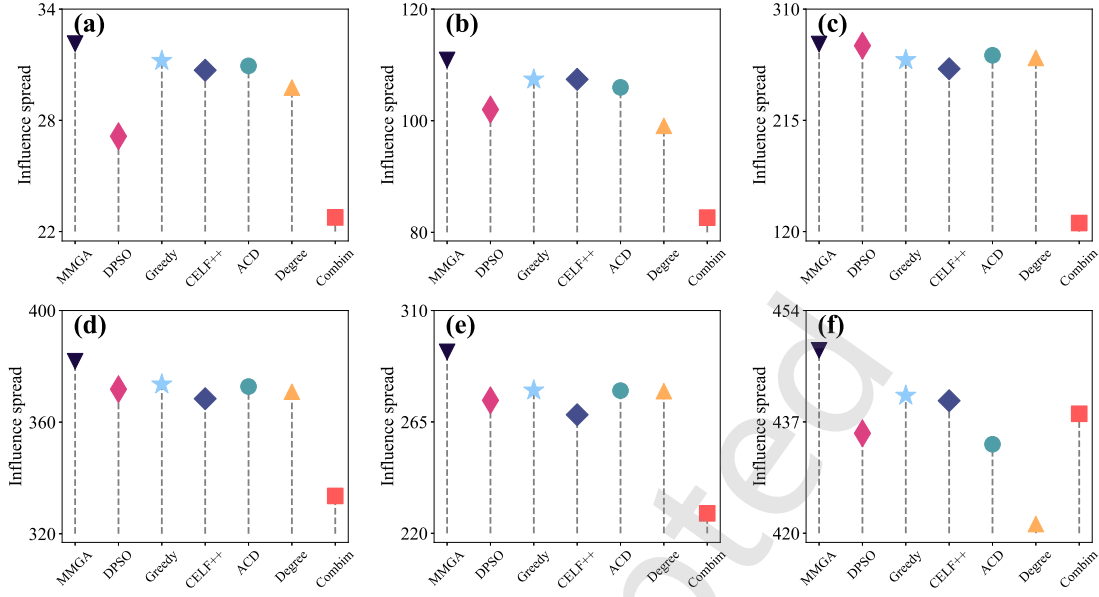
To verify the effectiveness of the repair module in MMGA, we conduct ablation experiments focusing on the repair module. In MMGA (without repair), if an individual exceeding the budget is generated during the evolutionary process, it is randomly replaced with an individual that meets the constraint. The improvement in MMGA compared to MMGA (without repair) is presented in Table 4, with an improvement ranging from

1.05% to 3.4%. Overall speaking, the presence of the repair module significantly enhances the performance of the MMGA algorithm.

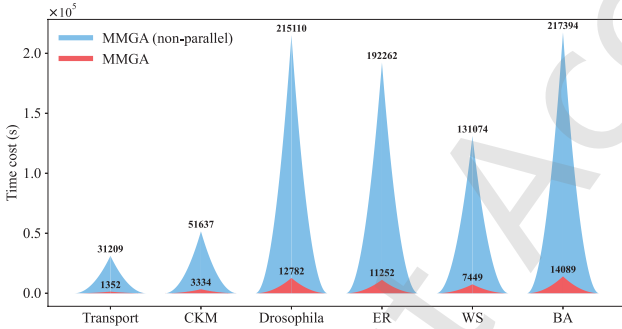
Furthermore, the significance of the parallel module in MMGA was further evaluated. The parallel module in MMGA is designed to reduce the time cost of the algorithm. Therefore, we assess its impact on computational efficiency. As illustrated in Figure 12, the runtime of the non-parallel implementation of MMGA is significantly greater than that of the parallel version, particularly on large multilayer networks.

#### 5.4 Complexity analysis

Additionally, we provide a detailed analysis of the time complexity of the MMGA algorithm. For a given



**Fig. 11** The influence spread achieved by different algorithms under randomly assigned node costs is evaluated, in which we set  $K = 20$  and  $B = 600$ . The results are given for networks: (a) Transport; (b) CKM; (c) Drosophila; (d) ER; (e) WS; and (f) BA.



**Fig. 12** The comparison of time costs between MMGA and MMGA (non-parallel) is presented, with parameters set to  $K = 20$  and  $B = 600$ .

population size  $K$  and individual size  $n$ , the time complexity of the Initialization and Repair functions is  $O(n \cdot K)$ . The Crossover and Mutation functions have a time complexity of  $O(R \cdot K)$ , while the selection process has a time complexity of  $O(K \cdot \log K)$ . Consequently, the overall time complexity of the MMGA algorithm is  $O(K \cdot (n + T \cdot (R + \log K)))$ .

## 6 Conclusion

Given that the current budgeted influence maximization problem only considers the limitation of budget, which may result in unreasonable solutions that are not applicable to real-world scenarios, we propose a new influence maximization problem in multilayer networks that considers both the budget and seed size, i.e., BCIM

problem. The challenges for solving this problem are mainly as follows: (i) How to design an algorithm that meets the two constraints in BCIM? (ii) How to find a seed set that is influential in different layers? To this end, we propose a multilayer multi-population genetic algorithm (MMGA), which uses repair and multi-population parallel modules to solve the budget and seed size constraints, respectively. In addition, the design of the initialization module and the fitness function in MMGA that considers nodes' properties and influence in multilayer networks can not only help to find the optimal influential seed set, but also navigate the algorithm to converge quickly. Meanwhile, a spreading model, i.e., MIC, which considers a piece of information spreading through a multilayer network and enhances inter-layer spread interactions, is proposed to quantify the spread influence of the nodes selected by different methods. To validate the effectiveness of our algorithm, we perform experiments on three artificial and three empirical multilayer networks from different domains. The results show that the proposed algorithm is superior to state-of-the-art benchmarks in terms of influence spread and robustness. Furthermore, through a detailed parameter analysis, we found that the crossover module and prior knowledge of initialization play crucial roles in the MMGA framework.

This paper presents the first attempt at solving the BCIM problem in multilayer networks and opens up

**Table 3** The AUC value under different initializations. The best performance is shown in bold and the second best is shown with an underline in each capacity constrain  $K$ .

Dataset	Initialization	K=10	K=12	K=14	K=16	K=18	K=20
Transport	Mixed	<b>0.2582</b>	<b>0.2569</b>	<b>0.2548</b>	<b>0.2532</b>	<b>0.2543</b>	<b>0.2548</b>
	Degree	<u>0.2470</u>	0.2473	0.2469	0.2489	0.2483	0.2475
	ICR	0.2464	<u>0.2466</u>	<u>0.2486</u>	0.2484	0.2478	0.2490
	Random	0.2484	<u>0.2492</u>	<u>0.2496</u>	<u>0.2495</u>	<u>0.2496</u>	<u>0.2487</u>
CKM	Mixed	<b>0.2509</b>	<b>0.2519</b>	<b>0.2524</b>	<b>0.2519</b>	<b>0.2527</b>	<b>0.2512</b>
	Degree	<u>0.2506</u>	0.2495	0.2490	0.2512	0.2447	0.2498
	ICR	0.2487	0.2483	0.2484	<u>0.2515</u>	<u>0.2513</u>	0.2490
	Random	0.2499	<u>0.2503</u>	<u>0.2502</u>	0.2454	<u>0.2513</u>	<u>0.2500</u>
Drosophila	Mixed	<b>0.2536</b>	<b>0.2531</b>	<b>0.2517</b>	<b>0.2503</b>	<b>0.2513</b>	<b>0.2505</b>
	Degree	0.2475	0.2481	0.2495	0.2501	0.2482	0.2498
	ICR	0.2488	<u>0.2500</u>	0.2489	0.2494	0.2499	0.2496
	Random	<u>0.2501</u>	0.2489	<u>0.2499</u>	<u>0.2502</u>	<u>0.2506</u>	<u>0.2501</u>
ER	Mixed	<b>0.2535</b>	<b>0.2545</b>	<b>0.2533</b>	<b>0.2527</b>	<b>0.2519</b>	<b>0.2516</b>
	Degree	0.2487	<u>0.2492</u>	0.2487	0.2491	0.2491	<u>0.2493</u>
	ICR	0.2478	0.2478	0.2483	0.2487	0.2491	0.2500
	Random	<u>0.2501</u>	0.2485	<u>0.2497</u>	<u>0.2496</u>	<u>0.2500</u>	0.2491
WS	Mixed	<b>0.2588</b>	<b>0.2584</b>	<b>0.2578</b>	<b>0.2570</b>	<b>0.2557</b>	<b>0.2552</b>
	Degree	0.2452	0.2450	0.2458	0.2479	0.2485	0.2469
	ICR	0.2466	0.2478	0.2469	0.2460	0.2468	0.2478
	Random	<u>0.2495</u>	<u>0.2488</u>	<u>0.2495</u>	<u>0.2491</u>	<u>0.2490</u>	<u>0.2501</u>
BA	Mixed	<b>0.2503</b>	<b>0.2501</b>	<b>0.2505</b>	<b>0.2506</b>	<b>0.2501</b>	<b>0.2506</b>
	Degree	<u>0.2500</u>	<u>0.2500</u>	0.2496	0.2499	0.2500	0.2497
	ICR	0.2497	0.2499	0.2498	0.2494	0.2498	0.2498
	Random	<u>0.2500</u>	<u>0.2500</u>	<u>0.2501</u>	<u>0.2501</u>	<u>0.2501</u>	<u>0.2499</u>

a series of future research directions. One is to design efficient algorithms when the costs of nodes are assigned in a more complex manner rather than solely based on node degree. Secondly, the multilayer independent cascade model used in this work is a simplification of the real-world scenario, which is similar to the independent cascade model on an aggregated weighted network of the corresponding multilayer network<sup>[48, 49]</sup>. In future work, one could consider a more realistic case where the spreading processes are characterized by different models to solve the BCIM problem. Besides, the fitness score of each node in the MMGA framework is obtained by Monte Carlo simulations, which have a relatively high computational complexity, especially on large-scale networks. Therefore, how to estimate the influence spread of each node in a multilayer network could also be a promising direction for exploration. Finally, we plan to apply the MMGA algorithm to real-world digital marketing scenarios to gain valuable insight into promoter selection strategies. This application has the potential to improve resource allocation efficiency and increase overall returns on investment.

**Table 4** The comparison of influence spread between MMGA and MMGA without the repair module, with parameters set to  $K = 20$  and  $B = 600$ , is presented. The improvement values are highlighted in bold for each multilayer network.

Ablation (repair module)			
Dataset	MMGA(without repair)	MMGA	Improvement
Transport	30.94	31.99	<b>3.40%</b>
CKM	110.40	112.80	<b>2.17%</b>
Drosophila	267.08	269.89	<b>1.05%</b>
ER	380.30	387.69	<b>1.94%</b>
WS	287.76	294.94	<b>2.50%</b>
BA	448.09	452.86	<b>1.06%</b>

## 7 Acknowledgment

This work was supported by the China Postdoctoral Science Foundation (2024M762809), the National Natural Science Foundation of China (Grant No. 62473123), the National Natural Science Foundation of China (Grant No. 62072115), Shanghai Science and Technology Innovation Action Plan Project (No. 22510713600).

## References

- [1] D. Notarmuzi, C. Castellano, A. Flammini, et al, Universality, criticality and complexity of information propagation in social media, *Nat. Commun.*, vol. 13, no. 1, pp. 1-8, 2022.
- [2] P. Huang, L. Guo, Y. Zhong, Efficient algorithms for maximizing group influence in social networks, *Tsinghua Sci. Technol.*, vol. 27, no. 5, pp. 832–842, 2022.
- [3] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, D.C, USA, 2003.
- [4] T. Parmer, L. M. Rocha, F. Radicchi, Influence maximization in boolean networks, *Nat. Commun.*, vol. 13, no. 1, pp. 1-11, 2022.
- [5] J. Wu, N. Wang, Approximating special social influence maximization problems, *Tsinghua Sci. Technol.*, vol. 25, no. 6, pp. 703–711, 2020.
- [6] N. Neophytou, A. Taik, G. Farnadi, Promoting fair vaccination strategies through influence maximization: A case study on covid-19 spread, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 85–93, 2024.
- [7] S. Kumar, A. Mallik, A. Khetarpal, B. S. Panda, Influence maximization in social networks using graph embedding and graph neural network, *Inf. Sci.*, vol. 607, pp. 1617–163, 2022.

- [8] S. Yao, N. Fan, J. Hu, Modeling the spread of infectious diseases through influence maximization, *Optim. Lett.*, vol. 16, no. 5, 1563–1586, 2022.
- [9] A. Goyal, W. Lu, L. V. S. Lakshmanan. Celf++ optimizing the greedy algorithm for influence maximization in social networks. *Proceedings of the 20th international conference companion on World wide web*, pp. 47-48, 2011.
- [10] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, USA, 2007.
- [11] G. Rao, D. Li, Y. Wang, et al. Maximizing the influence with k-grouping constraint[J]. *Inf. Sci.*, vol. 629, pp. 204-221, 2023.
- [12] R. Yan, H. Du, Y. Li, et al. Target users' activation probability maximization with different seed set constraints in social networks[J]. *Theor. Comput. Sci.*, vol. 838, pp. 111-125, 2020.
- [13] F. Morone, B. Min, L. Bo, et al. Collective influence algorithm to find influencers via optimal percolation in massively large social media[J]. *Sci. Rep.*, vol.6, no. 1, pp. 1-11, 2016.
- [14] W. Chen, Y. Wang, S. Yang. Efficient influence maximization in social networks[C]// *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199-208, 2009.
- [15] Y. Wang, G. Cong, G. Song, et al. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks[C]// *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 39-48, 2010.
- [16] H. Huang, H. Shen, Z. Meng, et al. Community-based influence maximization for viral marketing. *Appl. Intell.*, vol.49, pp. 37-50, 2019.
- [17] S. Bian, Q. Guo, S. Wang, J. X. Yu, Efficient algorithms for budgeted influence maximization on massive social networks, *Proc. VLDB Endow.*, vol. 13, no. 9, pp. 1498–1510, 2020.
- [18] S. Han, F. Zhuang, Q. He, Z. Shi, Balanced seed selection for budgeted influence maximization in social networks, *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference*, Tainan, China, 2014.
- [19] C. Wang, J. Zhao, L. Li, L. Jiao, J. Liu, K. Wu, A multi-transformation evolutionary framework for influence maximization in social networks, *IEEE Comput. Intell. Mag.*, vol. 18, no. 1, pp. 52–67, 2023.
- [20] X. Xu, Y. Zhang, Q. Hu, C. Xing, A balanced method for budgeted influence maximization, *SEKE*, pp. 250-255, 2015.
- [21] H. Nguyen, R. Zheng, On budgeted influence maximization in social networks, *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1084–1094, 2013.
- [22] R. De Souza, D. R. Figueiredo, A. d. A. Rocha, A. Ziviani, Efficient network seeding under variable node cost and limited budget for social networks, *Inf. Sci.*, vol. 514, pp. 369–384, 2020.
- [23] Y. Wang, W. Huang, L. Zong, T. Wang, D. Yang, Influence maximization with limit cost in social network, *Sci. China Inf. Sci.*, vol. 56, pp. 1-14, 2013.
- [24] C. V. Pham, H. V. Duong, H. X. Hoang, M. T. Thai, Competitive influence maximization within time and budget constraints in online social networks: An algorithmic approach, *Appl. Sci.*, vol. 9, no. 11, pp. 1-29, 2019.
- [25] J. Chen, Y. Deng, Z. Su, S. Wang, C. Gao, X. Li, Identifying multiple influential users based on the overlapping influence in multiplex networks, *IEEE Access*, vol. 7, pp. 50–59, 2019.
- [26] A. Kuhnle, M. A. Alim, X. Li, H. Zhang, M. T. Thai, Multiplex influence maximization in online social networks with heterogeneous diffusion models, *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 2, pp. 418–429, 2018.
- [27] K. V. Rao, C. R. Chowdary, Cbim: Community-based influence maximization in multilayer networks, *Inf. Sci.*, vol. 609, pp. 578–594, 2022.
- [28] C. R. Chowdary, et al., K++ shell: Influence maximization in multilayer networks using community detection, *Comput. Netw.*, vol. 234, pp. 1-9, 2023.
- [29] S. Wang, X. Tan, Solving the robust influence maximization problem on multi-layer networks via a memetic algorithm, *Appl. Soft Comput.*, vol. 121, pp. 1-15, 2022.
- [30] Z. Yuan, M. Shao, Z. Chen, Graph bayesian optimization for multiplex influence maximization, *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, British, 2024, pp. 75–83.
- [31] S. Wang, J. Liu, Y. Jin, Finding influential nodes in multiplex networks using a memetic algorithm, *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 900–912, 2019.
- [32] M. Katukuri, M. Jagarapu, et al., Cim: clique-based heuristic for finding influential nodes in multilayer networks, *Appl. Intell.*, vol. 52, no. 5, pp. 5173–5184, 2022.
- [33] X. M. Hu, Y. Q. Zhao, Z. Yang, Nodes grouping genetic algorithm for influence maximization in multiplex social networks, *Proceedings of the 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Rio de Janeiro, Brazil, 2023, pp. 1130–1135.
- [34] M. De Domenico, More is different in real-world multilayer networks, *Nat. Phys.*, vol. 19, no. 9, pp. 1247–1262, 2023.
- [35] Z. Xu, S. S. Chopra, Interconnectedness enhances network resilience of multimodal public transportation systems for safe-to-fail urban mobility, *Nat. Commun.*, vol. 14, no. 1, pp. 1-11, 2023.
- [36] L. Zhang, Y. Liu, F. Cheng, et al., A local-global influence indicator based constrained evolutionary algorithm for budgeted influence maximization in social networks, *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1557–1570, 2021.
- [37] X. L. Ren, N. Gleinig, D. Helbing, et al., Generalized network dismantling, *Proc. Natl. Acad. Sci.*, vol. 116, no. 14, pp. 54–59, 2019.

- [38] M. Gong, J. Yan, B. Shen, L. Ma, Q. Cai, Influence maximization in social networks based on discrete particle swarm optimization, *Inf. Sci.*, vol. 367, pp. 600–614, 2016.
- [39] S. S. Zhang, M. Xie, C. Liu, X. X. Zhan, Influence maximization in multilayer networks based on adaptive coupling degree, arXiv preprint arXiv:2311.08663, 2023.
- [40] S. Banerjee, M. Jenamani, D. K. Pratihar, Combim: A community-based solution approach for the budgeted influence maximization problem, *Expert Syst. Appl.*, vol. 125, pp. 1–13, 2019.
- [41] Y. Pan, D. H. Li, J. G. Liu, J. Z. Liang, Detecting community structure in complex networks via node similarity, *Physica A*, vol. 389, no. 14, pp. 2849–2857, 2010.
- [42] G. Caldarelli, E. Arcaute, M. Barthelemy, M. Batty, C. Gershenson, D. Helbing, S. Mancuso, Y. Moreno, J. J. Ramasco, C. Rozenblat, et al., The role of complexity for digital twins of cities, *Nat. Comput. Sci.*, vol. 3, no. 5, pp. 374–381, 2023.
- [43] M. De Domenico, A. Solé-Ribalta, S. Gómez, A. Arenas, Navigability of interconnected networks under random failures, *Proc. Natl. Acad. Sci.*, vol. 111, no. 23, pp. 8351–8356, 2014.
- [44] Albert R. Barabási A L, Statistical mechanics of complex networks. *Rev. Mod. Phys.*, vol. 74, no. 1, pp. 1-54, 2002.
- [45] M. De Domenico, A. Solé-Ribalta, S. Gómez, A. Arenas, Navigability of interconnected networks under random failures, *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 51–56, 2014.
- [46] J. Coleman, E. Katz, H. Menzel, The diffusion of an innovation among physicians, *Sociometry*, vol. 20, no. 4, pp. 253–270, 1957.
- [47] C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, M. Tyers, Biogrid: a general repository for interaction datasets, *Nucleic Acids Res.*, vol. 34, pp. 535–539, 2006.
- [48] H. Wang, J. M. Hernandez, P. Van Mieghem, Betweenness centrality in a weighted network, *Phys. Rev. E*, vol. 77, no. 4, pp. 1-10, 2008.
- [49] J. Xie, X. Wang, Y. Liu, et al. Social Media-Driven User Community Finding with Privacy Protection. *Tsinghua Sci. Technol.*, vol. 30, no. 4, pp. 82-92, 2025.



**Su-Su Zhang** received the bachelor's degree in software engineering from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2022. He is currently pursuing the master's degree with the Alibaba Research Center for Complexity Sciences, Hangzhou Normal University, Hangzhou, China.

His current research interests include complex networks, data mining, and social computing.



**Chuang Liu** received his Ph. D. degree in Process System Engineering in 2012 from East China University of Science and Technology, Shanghai, China. He is a professor in the Alibaba Research Center for Complexity Sciences at the Hangzhou Normal University, Hangzhou, China. His current research interests are mainly focused on network science and data mining.



**Huijuan Wang** received the M.Sc. and Ph.D. degrees in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 2005 and 2009, respectively. Since 2011, she has been a Visiting Scientist with the Department of Physics, Boston University, Boston, MA, USA, and in 2015, with the Department of Electrical Engineering, Stanford University, Stanford, CA, USA. She is currently an Associate Professor with the Multimedia Computing Group, Department of Intelligent Systems, Delft University of Technology. Her main research interests include prediction, modeling and control of dynamic processes (such as information, disease, failure propagation and social or financial contagion) on interdependent and time-evolving networks combining data and network science approaches.



**Yang Chen** received the BS and PhD degrees from the Department of Electronic Engineering, Tsinghua University, in 2004 and 2009, respectively. He is a professor at the School of Computer Science, Fudan University, China. He leads the Big Data and Networking (DataNET) Group since 2014. Before joining Fudan, he was a postdoctoral associate with the Department of Computer Science, Duke University, USA, where he served as a senior personnel with the NSF MobilityFirst project. From September 2009 to April 2011, he has been a research associate and the deputy head of Computer Networks Group, Institute of Computer Science, University of Göttingen, Germany. He visited Stanford University (in 2007) and Microsoft Research Asia (2006-2008) as a visiting student. His research interests include online social networks, Internet architecture and applied machine learning. He serves as an associate editor-in-chief of the Journal of Social Computing. He is a senior member of ACM, IEEE, and CCF.



**Xiu-Xiu Zhan** received the Ph.D. degree in computer science from the Delft University of Technology, Delft, The Netherlands, in 2020. She is currently an Associate Professor at Alibaba Business School, Hangzhou Normal University, Hangzhou, China. Her research interests include complex networks, information diffusion, and machine learning. She has authored or co-authored

more than 40 international journal papers and conference papers in physics and computer science, including Physics Reports,

Information Processing & Management, and IEEE Transactions on Network Science and Engineering.

Just Accepted