



ARTICLE

# A Hybrid Clique-Based Method with Structural Feature Node Extraction for Community Detection in Overlapping Networks

Sicheng Ma<sup>1</sup>, Lixiang Zhang<sup>2,\*</sup>, Guocai Chen<sup>3</sup>, Zeyu Dai<sup>3</sup>, Junru Zhu<sup>4</sup> and Wei Fang<sup>1,\*</sup>

<sup>1</sup>School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China

<sup>2</sup>School of Continuing Education, Jiangnan University, Wuxi, China

<sup>3</sup>China Ship Scientific Research Center, Wuxi, China

<sup>4</sup>Amazon-Alexa Shopping, Seattle, WA, USA

\*Corresponding Authors: Lixiang Zhang. Email: 9202400446@jiangnan.edu.cn; Wei Fang. Email: fangwei@jiangnan.edu.cn

Received: 21 September 2025; Accepted: 29 December 2025; Published: 10 February 2026

**ABSTRACT:** Community detection is a fundamental problem in network analysis for identifying densely connected node clusters, with successful applications in diverse fields like social networks, recommendation systems, biology, and cyberattack detection. Overlapping community detection refers to the case of a node belonging to multiple communities simultaneously, which is a much more meaningful and challenging task. Graph representation learning with Evolutionary Computation has been studied well in overlapping community detection to deal with complex network structures and characteristics. However, most of them focus on searching the entire solution space, which can be inefficient and lead to inadequate results. To overcome the problem, a structural feature node extraction method is first proposed that can effectively map a network into a structural embedding space. Thus, nodes within the network are classified into hierarchical levels based on their structural feature strength, and only nodes with relatively high strength are considered in subsequent search steps to reduce the search space. Then, a maximal-clique representation method is employed on the given vertex set to identify overlapping nodes. A hybrid clique-based multi-objective evolutionary algorithm with decomposition method is designed to address cliques and the remaining unexplored nodes separately. The number of communities generated with this allocation method is closer to the actual partition count with high division quality. Experimental results on nine usually used real-world networks, five synthetic networks, and two large-scale networks demonstrate the effectiveness of the proposed methodology in terms of community quality and algorithmic efficiency, compared to traditional, MOEA-based, and graph embedding-based community detection algorithms.

**KEYWORDS:** Community detection; graph embedding; multi-objective evolutionary algorithm; cliques; link strength

## 1 Introduction

Community detection is used to identify groups of nodes that have similar properties or functions, which is a challenging problem in network analysis [1]. Conventional community detection methods need comprehensive network information, including connectivity, properties, or labels. However, traditional methods, such as the Girvan-Newman algorithm [2], Louvain algorithm [3], and Infomap algorithm [4], are insufficient for community detection in networks characterized by complex structures or dynamic properties [5]. These traditional methods generate a large number of small or poorly defined communities, which can be difficult to interpret and not be useful in downstream analyses [6].

An overlapping network is a type of network in which nodes can belong to more than one community or group simultaneously, such as social networks, biological networks, and technological networks [7,8]. In social networks, for example, individuals may belong to multiple communities according to their interests, professions, or social connections [9]. In recent years, there has been a growing interest in the development of algorithms employing probabilistic modeling [10], graph partitioning [11], and deep learning [12].

To further improve the quality of community detection, the development of overlapping community detection methods based on multi-objective evolutionary algorithms (MOEA) has become an increasingly active research area [13–16]. Earlier MOEA-based methods typically optimized only an objective function, and thus had difficulty accurately capturing multiple community structures within a network [17]. Therefore, to address the increasing diversity of optimization problems, the use of multiple objective functions for community detection has been proposed. Multi-objective methods consider modularity, community density, and coverage, to enhance the global quality of the detected communities. However, MOEA-based algorithms exhibit high computational complexity, especially when dealing with large-scale networks, resulting in significantly increased computation times. The existing community detection methods are sensitive to the assumptions of the network representation and the model, which limits their applicability to specific contexts [8]. For example, MOGA-Net [13] is a multi-objective algorithm that employs locus-based representation, to identify high-quality communities that accurately reflect the underlying structure of a network. MOEA/D-Net [17] uses a decomposition-based approach to identify communities in networks, based on the principles of multi-objective optimization. Wen et al. [18] designed a maximal clique-based representation to identify highly overlapping communities by finding all maximal cliques in the network, which are complete subgraphs that cannot be extended by adding more nodes. The aforementioned methods are restricted by their different encoding and decoding techniques [19].

In recent years, many research papers have proposed several kinds of deep learning approaches for community detection due to their strong ability to learn nonlinear network properties, preserve complicated network structures with lower dimensional network embeddings, and detect communities more accurately [20]. Representative deep learning-based community detection approaches include Convolutional Network-based [21], Graph Convolutional Network(GCN)-based [22], Graph Attention Network-based [23], Generative Adversarial Network-based [24], Autoencoder-based [7,25], Deep Sparse Filtering-based [26], and Deep Nonnegative Matrix Factorization-based [27]. Among them, GCN-based community detection has achieved great interest as it can aggregate node neighborhood information in deep graph convolutional layers to globally capture complex features [28]. GCNs provide a powerful, flexible, and modern deep learning framework for generating high-quality, inductive graph embeddings. Graph embedding-based community detection methods have emerged as a promising alternative to traditional methods [29]. The graph embedding-based community detection methods aim to transform network data into a low-dimensional continuous vector space, enabling more effective community detection in complex networks. Existing graph embedding-based methods, such as DeepWalk [30], Node2vec [31], and LINE [32], which are shallow embedding, cannot always produce high-quality embeddings or provide satisfactory results for community detection. For example, the DeepWalk and Node2vec algorithms rely on random walks to generate node sequences, which may not accurately capture the global network structure. In addition, graph embedding methods are often limited by the quality and quantity of available network data, which can affect the accuracy and reliability of community detection results [33]. In the last two years, an increasing amount of research on graph decomposition-based community detection methods emerged. The CPGC method [22] leverages the idea of graph decomposition by modifying the graph convolution to combine representation learning and clustering, achieving improved community partition results. In the HAE algorithm [34], the non-negative factorization of the matrix is used to decompose the characteristics

of the network, using the semantics obtained to assist in training an unsupervised community detection algorithm. These methods have validated the applicability of the graph decomposition approach in the field of community detection.

Based on the aforementioned intuitions, in the paper, a novel structural feature-based maximal clique MOEA/D algorithm, SFCMOEA, is proposed for the problem of overlapping community detection in networks. In SFCMOEA, a series of methods are presented that combine graph embedding-based methods, the maximal clique representation scheme, and the decomposition-based MOEA(MOEA/D) framework. The main contribution of this article can be summarized as follows.

- (1) A structural node extraction method is proposed, where the graph embedding method [35] and the Minkowski distance calculation [36] are used to represent the structure of the network and the relationships between the nodes.
- (2) A hybrid clique-based MOEA/D (HCMOEA) method is designed to optimize multiple objectives, with the aim of obtaining high-quality community partitioning results.
- (3) Experiments are conducted on a series of synthetic and real-world networks of varying scales to demonstrate the usability and reliability of the SFCMOEA algorithm in addressing the community detection problem.

The remainder of the paper is organized as follows. [Section 2](#) briefly introduces community detection problems and the framework of MOEA-based community detection methods. The SFCMOEA and its core components, including the SFNE method, the maximal-clique-based representation scheme, and the HCMOEA, are described in detail in [Section 3](#). In [Section 4](#), the performance of the proposed SFCMOEA is evaluated on synthetic and real-world networks, compared to six representative methods. Finally, the conclusions are presented in [Section 5](#).

## 2 Related Work

In this section, a more detailed explanation of concepts related to overlapping networks and the application of community detection algorithms is presented. Then the MOEA model of the community detection problem is given. Finally, the MOEA framework and the objective functions used are briefly reviewed.

### 2.1 Community Detection for Overlapping Networks

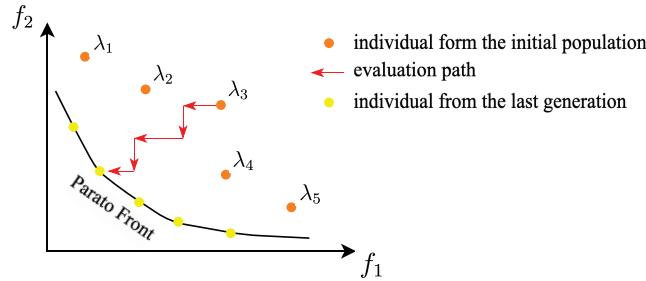
Overlapping networks are a type of complex network in which nodes can belong to multiple communities simultaneously, which often exhibit multiple affiliations or memberships [37]. Mathematically, an overlapping network can be represented as a graph  $G(V, E)$ , where  $V$  denotes the set of nodes and  $E$  denotes the set of edges between nodes. The overlapping community structure is defined as a collection of communities  $\{C_1, C_2, \dots, C_k\}$ , where  $C_i \subseteq V$  and  $C_i \cap C_j \neq \emptyset$  for at least one pair of communities  $(i, j)$  [37].

Community detection is the process of identifying groups of nodes that are densely connected within the group while having relatively few connections to nodes outside that group. In the context of overlapping networks, the goal is to identify communities such that each node can belong to one or more communities, depending on its connections and attributes. Overlapping community detection methods aim to identify overlapping communities within networks. Traditional methods, such as the clique percolation method (CPM) [38], and the local expansion and optimization method (LEOM) [39], are based on connections or attribute relationships among nodes in the network. These methods have some advantages in terms of interpretability and performance, but often suffer from scalability issues and are sensitive to network

structures and parameter settings. In recent years, the Node Importance-Based Label Propagation Algorithm (NI-LPA) has been proposed to uncover logical partitions on real networks [40].

## 2.2 MOEA/D

MOEA/D is effectively applied to community detection by formulating the problem as a multi-objective optimization problem (MOP). For community detection problem, optimizations targeting such as maximizing modularity, minimizing intercommunity edges, and ensuring balanced community sizes. The decomposition-based method, MOEA/D, allows simultaneous optimization of multiple subproblems. A simple illustration of MOEA/D is given in Fig. 1. In the figure, the orange nodes represent the initial population. Each orange point corresponding to the individual solutions evaluated according to two objectives,  $f_1$  and  $f_2$ . The yellow nodes connected by a curve represent the Pareto front, which signifies the set of non-optimal solutions. The selected node  $\lambda_3$  reaches the Pareto front through several evolutionary iterations. Each segment of the red line represents the individual's optimization process for either  $f_1$  or  $f_2$ .



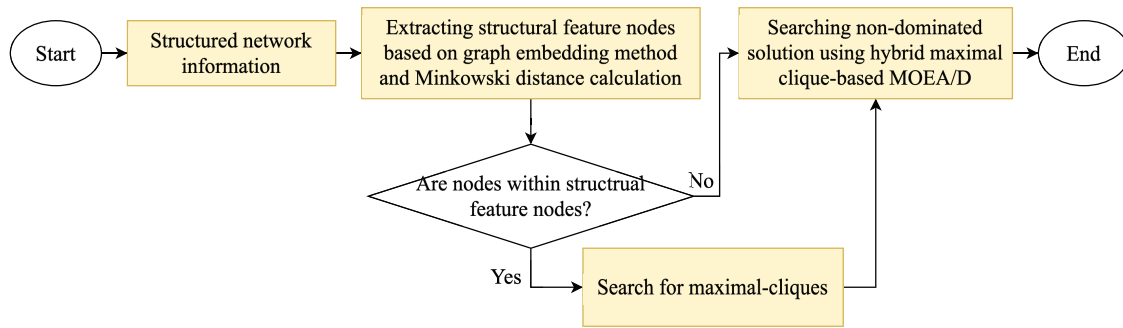
**Figure 1:** A simple illustration of the optimization process in MOEA/D

In the development of MOEA/D, Zhu et al. [41] used a locus-based representation in conjunction with the multiobjective community detection (MOCD) method to improve the quality of community partition results. Furthermore, Zhang et al. [42] introduced the concepts of macro- and micropopulation representations to obtain robust preliminary community structures. The framework of the MOEA-based community detection method is introduced. The common characteristic shared by the two methods mentioned above and earlier MOEA-based approaches is the presence of a representation step. In many studies, the representation step aims to retain the most important structural and topological characteristics of the network [17]. In order to better compatibility with MOEA-based techniques, complex networks can be transformed into more analytical forms, by means of dimensionality reduction, noise reduction, feature extraction, or conversion of network data into formats. Representation as a type of preprocessing step is important in improving the quality of network data and optimizing inputs for MOEAs [18].

In the MOEA framework, the selection of both the representation scheme and the objective functions is of significant importance. One of the chosen objective functions is the kernel K-means (KKM) function, which takes advantage of kernel functions to map network data to a higher-dimensional feature space. On the other hand, Ratio Cut (RC) is employed as another objective function because of its focus on achieving balanced and well-separated communities. A lower KKM value indicates a higher intralink density, while a lower RC value signifies a lower interlink density. The community detection problem can be formulated as a two-objective optimization problem by minimizing the KKM and RC objectives simultaneously [43].

### 3 Structural Feature Clique-Based MOEA

This section provides a comprehensive introduction to the proposed methodology, and the flowchart for the proposed method, SFCMOEA, is presented in Fig. 2. As can be seen in Fig. 2, there are mainly four steps in SFCMOEA, which are generating structured network information, extracting structural feature nodes, searching for maximal-cliques, and searching for non-dominated solution using hybrid maximal clique-based MOEA/D. For extracting structural feature nodes, graph embedding and Minkowski distance calculation are employed to represent the structure and relationships of nodes, thereby better revealing hidden community structures and preserving the network structure in the similarity matrix. A series of nonlinear functions is applied to the matrix to identify a set of structural feature nodes, where each node can represent a potential partition. Subsequently, an improved maximal clique search method is employed to identify all maximal cliques based on its efficient search capability. In the HCMOEA, the populations formed by maximal cliques and nonclique nodes are processed separately. Individuals represented by maximal cliques are optimized, while non-clique nodes are allocated based on connection strength to existing communities. In SFCMOEA, nodes belonging to maximal cliques and those that do not are processed separately, which can not only improve search efficiency, but also achieve more accurate partitioning for non-clique nodes by incorporating the calculation of neighborhood similarity.



**Figure 2:** The flowchart of the proposed algorithm SFCMOEA

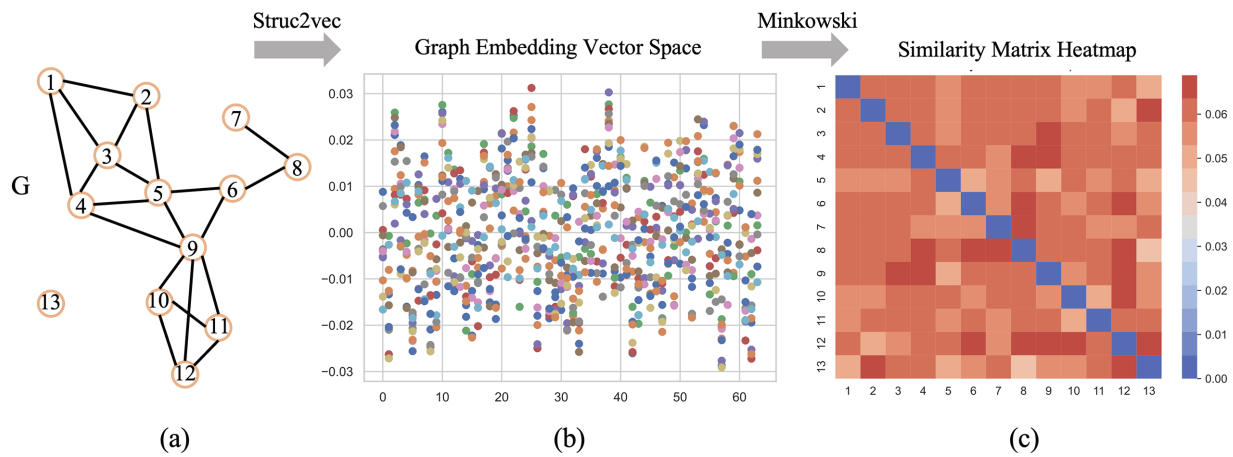
#### 3.1 Structural Feature Node Extraction (SFNE) Method

The search process in an overlapping network space is often affected by isolated nodes and hidden communities, leading to a decrease in search efficiency. To overcome this problem, the challenge is to map the high-dimensional network space onto a lower-dimensional space while preserving the information about the network structure. In addition, it is important to maintain the consistency of the network structure during subsequent search processes and to effectively handle isolated nodes and hidden communities. Thus, a method that can accurately preserve the network structure while considering isolated community structures is needed. A method named structural feature node extraction (SFNE) is proposed, as presented in Algorithm 1, which encodes connectivity relationships between nodes using graph embedding and similarity computation methods. The input to Algorithm 1 is a structural embedding matrix  $\phi$ , obtained by the struc2vec algorithm [44]. To further measure the connections between the nodes, the Minkowski distance formula [36] is used, which is expressed as:

$$d_{ij} = \left( \sum_{k=1}^n |v_{ik} - v_{jk}|^p \right)^{\frac{1}{p}} \quad (1)$$

where  $d_{ij}$  represents the Minkowski distance between nodes  $i$  and  $j$ . Here,  $v_{ik}$  and  $v_{jk}$  denote the  $k$ -th feature values of nodes  $i$  and  $j$ , respectively. The parameter  $p$  determines the order of the Minkowski distance.

To more directly illustrate the process of the SFNE method, the graph embedding process and the calculation of node connections based on the original graph  $G$  are presented in Fig. 3 from the original graph  $G$ . Fig. 3a represents a network with 13 nodes and 18 edges. By applying the struc2vec algorithm, a multidimensional embedding space is obtained, and each node is mapped to the embedding space, resulting in Fig. 3b. In the embedding space, each element describes the position and features of a node within the graph. Structurally similar nodes, with weaker colors, are embedded in similar positions. Fig. 3c shows the similarity matrix obtained by applying the Minkowski distance formula in the embedding space. More closely connected nodes tend to have higher weights, whereas isolated nodes are also assigned weights. In the graph, node 13 is an isolated node, while nodes  $\{3, 5, 9\}$  are closely connected. Nodes  $\{3, 5, 9\}$  tend to have higher weights and are more likely to be assigned to multiple communities, leading to the phenomena of overlap.



**Figure 3:** Illustration of the SFNE method, which includes transforming a network into embedding space vectors and then converting these vectors into a node similarity matrix. (a) Original network with 13 nodes and 18 links. (b) Space vectors calculated by the struc2vec algorithm, and different colors represent points of different dimensions. (c) The node similarity matrix, calculated using the Minkowski distance, represents each element as the position and features of a node within the graph, with darker colors indicating stronger structural features

---

#### Algorithm 1: Structural feature node extraction method

---

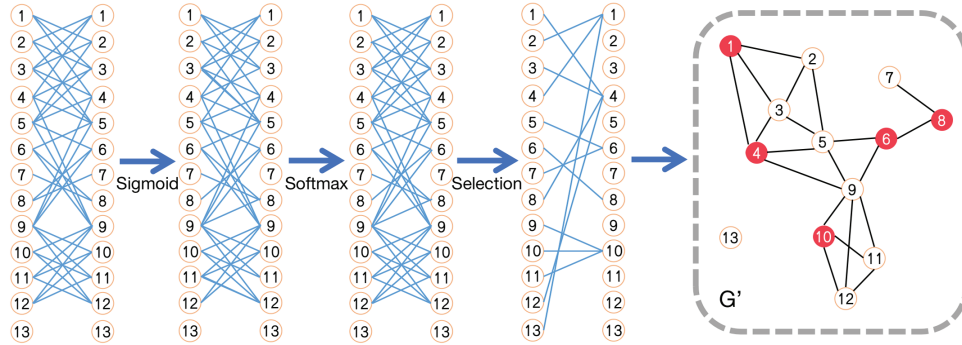
**Require:** Structural embedding matrix  $\phi$  from struc2vec.

**Ensure:** Structural feature node set  $\delta$ .

- 1:  $\delta = \emptyset$
  - 2:  $D \leftarrow \text{apply Eq. 1 to } \phi$
  - 3: **for**  $i = 1 \rightarrow N$  **do**
  - 4:    $h_i \leftarrow \sigma(D_i)$ ;
  - 5:    $p_i \leftarrow \text{softmax}(h_i)$ ;
  - 6:    $\text{sorted}_i \leftarrow \text{argsort}(p_i)$ ;
  - 7:   **for**  $j = 1 \rightarrow N\%1000$  **do**
  - 8:      $\delta \leftarrow x_{i, \text{sorted}_{i,j}}$ ;
  - 9:   **end for**
  - 10: **end for**
  - 11: **return**  $\delta$ .
-



On line 2 of Algorithm 1, the node distance matrix  $D$  is obtained by applying Eq. (1) to  $\phi$ . Then, from lines 3 to 9, the embedded features corresponding to each node are subjected to nonlinear transformations and normalization using the sigmoid and softmax functions. The nonlinear functions tend to transform a network into a continuous probability values. Subsequently, the nodes are sorted and selected based on the probability values, with the top  $N\%1000$  nodes forming the structural feature vertex set, denoted by  $\delta$ . As seen in Fig. 4, the labeled nodes in  $\delta$  represent tightly connected structural relationships in the network. The introduction of random numbers adds diversity to the generated vertex set. A search initiated from such a vertex set can cover a wider range of scenarios.



**Figure 4:** Application of the SFNE method to graph  $G$  in Fig. 3c, generating graph  $G'$ . And the structural feature nodes marked in red on  $G'$

Fig. 4 illustrates the procedure of the SFNE method, with graph  $G$  as an example. Applying the sigmoid and softmax functions on the matrix  $D$ , the probabilities of each node belonging to all other nodes are obtained. Afterwards, each node has  $v/1000 + 1$  opportunities to make a selection to choose which nodes to connect with. The set of vertex features of the structural feature is thus formed by the selected nodes to represent the structural features of the given network. In Fig. 4, nodes  $\{1, 4, 6, 8, 10\}$  compose the set of structural feature vertex. In addition, the nodes within the vertex set serve as the centers of the potential community structures. For example, nodes  $\{4, 6, 10\}$  have more connected nodes than the other nodes. The  $v_6$  can be seen as the bridge between two structures. At the same time, the isolated node  $v_{13}$  participates in the assignment, indicating a preference to be part of the community formed by the only node.

### 3.2 Representation Scheme

After obtaining the set of structural feature vertex, the next step is to apply the representation scheme. Taking into account both efficiency and search accuracy, the maximal-clique algorithm [18] is chosen here. A detailed description of the maximal clique algorithm is shown in Algorithm 2; on line 1, an empty set  $V^K$  is used as the candidate maximal clique, where  $K$  is the number of searched cliques.

---

#### Algorithm 2: Finding the maximal clique of $G$

---

**Require:**  $G(V, E)$

**Ensure:** Set of clique nodes  $V^K$

- 1:  $V^K = \emptyset$
  - 2:  $V^{\text{sorted}} \leftarrow \text{sort}(V, d)$
  - 3:  $\text{minSize} \leftarrow 3$
  - 4: **for**  $k \in d^{\text{sorted}}$  **do**
  - 5:   **for**  $\text{node} \leftarrow V^{\text{sorted}}$  **do**
- 

(Continued)

**Algorithm 2 (continued)**


---

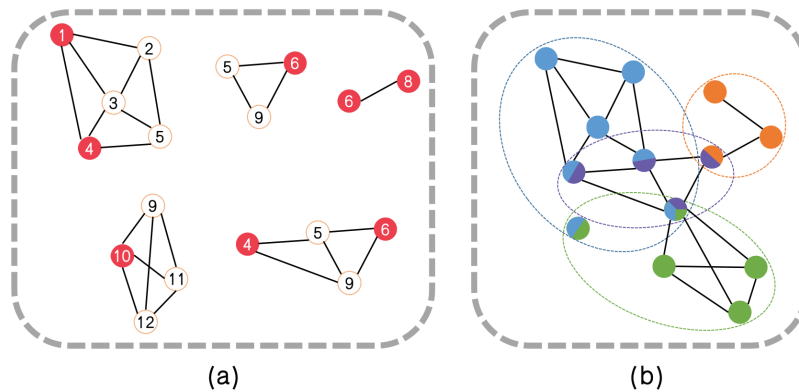
```

6:    $neigh \leftarrow neighbor(node)$ 
7:    $maxClique \leftarrow findClique(node)$ 
8:   if  $size(maxClique) > minSize$  then
9:      $V^K \leftarrow maxClique$ 
10:  end if
11: end for
12: end for
13: return  $V^{sorted}$ .

```

---

Then the unvisited nodes are each iteratively selected as the next candidate node in reverse order node degrees  $d$ . The selected node is added to the current candidate maximal-clique and checked for adjacency with all nodes not in the candidate set. Each node adjacent to all nodes in the maximal-clique candidate is also added to the candidate set, and the algorithm proceeds to the next step. If the selected node is not adjacent to any other nodes, then the algorithm backtracks to the previous node. The known maximal-clique is updated, when the current candidate maximal-clique is larger than the known maximal-clique. Lines 4 to 12 of the algorithm are repeated until all nodes have been visited. Using the maximal clique representation in our example  $G'$ , five cliques are obtained, as shown in Fig. 5a. In Fig. 5b, the final partitioning results are obtained through SFCMOEA with four distinct communities in four colors. There are four overlapping nodes, with one node belonging to a maximum of three communities. In the figure, almost all network structures are covered, indicating the effectiveness of the strategy of starting the search from the set of structural feature vertex. Furthermore, due to the characteristics of the maximal clique algorithm, the structures formed by nodes  $\{1, 2, 3, 4, 5\}$  and  $\{9, 10, 11, 12\}$  are particularly robust.



**Figure 5:** Example of the maximal cliques and final partitioning results. (a) The maximal cliques obtained by searching in  $G'$  starting from the red structural feature nodes. (b) The final partitioning results obtained through SFCMOEA with four distinct communities in four colors. Four overlapping nodes, with one node belonging to a maximum of three communities is presented

Compared with the MOEA step, preparatory work is performed to compute the weights to quantify the individuals in the clique structure. Once the clique are obtained, each clique in Fig. 5a is treated as an individual. The weights between individuals are calculated, and two cliques merge if the combined weight exceeds that of the individual cliques. The weight, also called the link strength, is defined on the basis of the



interrelation between the internal and external nodes.

$$density(C_i, C_j) = \frac{|(u, v) | u \in C_i, v \in C_j, (u, v) \in E|}{|C_i|} \quad (2)$$

As seen in Eq. (2), both the number of intercommunity edges and the average number of edges are utilized to assess *density*. Based on the definition of *density*, the strength of the link between communities  $C_i$  and  $C_j$  is defined as:

$$W(C_i, C_j) = \frac{density(C_i, C_j) \times density(C_j, C_i)}{|E^{in}C_i| \times |E^{in}C_j|} \quad (3)$$

The calculated link strength  $W$  is used to determine whether two cliques should be merged as follows:

$$T(c) = \frac{W_{in}(c)}{W_{in}(c) + W_{out}(c)} \quad (4)$$

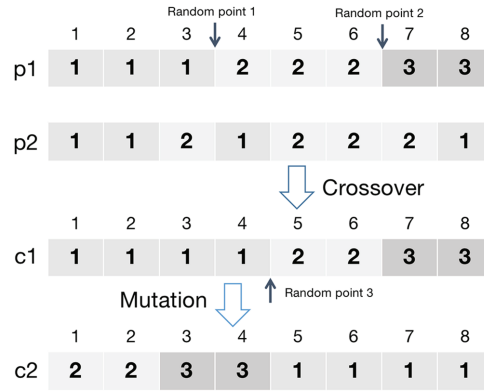
In the definition of  $T$ , the maximal clique graph is treated as a weighted network, where  $W_{in}$  and  $W_{out}$  represent the total weights of the internal and external edges, respectively, of the clique  $c = |C_i, C_j|$ . If  $T(c) > 0$ , then the two cliques are merged and their genotype is updated.

### 3.3 Hybrid Clique-Based MOEA

As described in the following subsection, the proposed hybrid clique-based MOEA/D (HCMOEA) method is specifically designed to operate on clique-structured individuals, providing advantages over traditional techniques in terms of maintaining structural integrity and enhancing diversity. As illustrated in Fig. 5a, each maximal-clique in a graph is a complete subgraph in which every node is connected to every other node. Considering the clique structure, we divide the algorithm into two parts: the clique-based optimization step and the community allocation method. The aim is to maximize the advantages offered by tightly connected links within the discovered cliques.

#### 3.3.1 Clique-Based Optimization

In the HCMOEA, the proposed crossover and mutation operations are designed to preserve the maximal-clique structure of individuals while effectively increasing diversity. Here, the two parental individuals  $p1$  and  $p2$  in Fig. 6 are taken as an example. According to the structure based on cliques, each gene indicates a clique and  $g_1, g_2, g_3$  belong to the same clique. Apparently, for the same clique  $C_3$ , the community label is “1” in  $p1$  but “2” in  $p2$ . Two points are randomly selected to perform gene crossover between the two parents, resulting in the generation of new offspring. It can be seen that the offspring generated from  $p_2$ , which originally had only two divisions, now also exhibits three divisions. The crossover operation ensures that the offspring gene sequences maintain the connections between cliques, since each node remains connected to every other node in the clique. In the gene shuffle mutation method, a random number decides to exchange of genes, or the cliques within the child objects. If the random number is less than the specified probability, an exchange operation takes place. Otherwise, two random positions are generated to determine the subsequences, and a reversal operation is performed on the genes.



**Figure 6:** Example of the clique-based optimization process

As shown in Fig. 6, a mutation point is selected from the offspring obtained in the crossover step. The genes on both sides of the mutation point, namely  $\{g_1, g_2, g_3, g_4\}$  and  $\{g_5, g_6, g_7, g_8\}$ , are exchanged, resulting in  $c_2$ . As a result, the clique structure in  $C_2$  exhibited a significantly different partitioning than that of the original configuration. The mutation operation in HCMOEA is designed to enhance diversity without compromising the maximal-clique structure that increased diversity. The cliques obtained by combining the SFNE method with the maximal-clique representation are internally densely connected and externally sparsely connected partitions. A global exchange strategy can enhance diversity while ensuring that the original structure is less likely to be lost. The detailed procedure is shown in Algorithm 3.

---

**Algorithm 3:** Hybrid clique based MOEA/D

---

**Require:** Population size:  $PS$ ; maximum generation:  $T$ ;

neighborhood size:  $t$ ; set of uniform weight vectors:

$\{\lambda^1, \lambda^2, \dots, \lambda^{PS}\}$

**Ensure:** Collection of node  $\delta_V$ .

1: Work out the  $t$  closest weight vectors to each weight vector;

2: Initialize population  $P = I_1, I_2, \dots, I_{PS}$ ;

3: **for**  $i = 1 \rightarrow PS$  **do**

4:  $B(i) = i_1, \dots, i_t$ , where  $\lambda^{i_1}, \dots, \lambda^{i_t}$  are the  $t$  closest weight vectors to  $\lambda^i$ ;

5: **end for**

6: **for**  $g = 1 \rightarrow T$  **do**

7: **for**  $i = 1 \rightarrow PS$  **do**

8: **if**  $\text{rand}() < p_c$  **then**

9:  $child \leftarrow$  randomly select two crossover points and combine the corresponding gene sequences from the parents;

10: **else**

11:  $child \leftarrow P_i$

12: **end if**

13:  $child \leftarrow \text{GeneShuffleMutation}(child)$ ;

14: Evaluate child with  $KKM$  and  $RC$  objectives;

15: **for** individual  $I_j \in B(i)$  **do**

16:  $\text{EnvironmentSelect}(I_j, child)$ ;

17: **end for**

---

(Continued)

**Algorithm 3 (continued)**


---

```

18:    $P \leftarrow \text{CommunityAllocationMethod}(P)$ 
19: end for
20: end for
21: return  $\delta_V$ .

```

---

Regarding the objective functions, the kernel K-means(KKM) [45] and Ratio Cut(RC) [46] function are chosen. The KKM and the RC values of community  $C$  are calculated as:

$$KKM = 2(N - n) - \sum_{i=1}^n \frac{W(C_i, C_i)}{|C_i|} \quad (5)$$

$$RC = \sum_{i=1}^n \frac{W(C_i, \bar{C}_i)}{|C_i|} \quad (6)$$

where  $|C_i|$  denotes the size of a set  $C_i$ , and  $W(C_i, C_i)$  is the weight between cliques  $C_i$  and  $C_j$ .

### 3.3.2 Community Allocation Method

Through the clique-based optimization process, a series of internally densely connected structures with sparse external links are obtained. The next step is to assign the remaining nodes to the existing structures. The communities of isolated nodes and nodes within cliques with a size smaller than 3 are determined using the community allocation method. For example, in Fig. 5, the node  $n_{13}$  is isolated and did not participate in the optimization process. The measurements of these nodes with respect to the different cliques are calculated, and then each node is assigned to the clique with the highest weight. The measurement calculation is defined as follows:

$$S_{ij} = \frac{A_{i,j} + |\Gamma(i) \cap \Gamma(j)|}{(|\Gamma(i)| + |\Gamma(j)|)/2} \quad (7)$$

where  $A_{i,j}$  represents the attraction between nodes  $i$  and  $j$ , and  $\Gamma(i) \cap \Gamma(j)$  denotes the set of common neighbors of nodes  $n_i$  and  $n_j$ . The advantage of the allocation method is that it preserves the existing robust partition structure while avoiding an unnecessary increase in the number of partitions.

## 4 Experiments

In this section, the experimental results obtained by applying the proposed algorithm to real-world networks and synthetic benchmark networks are presented.

### 4.1 Experimental Networks

In Table 1, nine real-world networks with different numbers of nodes and links are presented. The smallest-scale network comprises only 34 nodes, whereas the largest-scale network comprises more than 2000 nodes. In addition, the ratio of nodes to edges varies significantly between the different networks. For the generation of synthetic networks, Lancichinetti Fortunato Radicchi (LFR) is a benchmark network model to evaluate community detection algorithms [47]. The LFR model is commonly used to evaluate community detection tasks because it can generate synthetic networks with known community structures [48]. The LFR model allows for adjustment of parameters such as the number of nodes, average node degree, maximum node degree, mixing parameter, community size distribution, and number of overlapping nodes to generate networks that simulate various real-world scenarios. Multiple parameters enable researchers to analyze

the performance of community detection methods under different conditions. Table 2 lists the various parameters considered:  $N$ ; the average node degree  $k$ ; the maximum node  $k_{max}$ ; the mixing parameter  $\mu \in [0, 1]$ ; the parameters of the community size distribution,  $t_1$ ,  $t_2$ ; and the number of overlapping nodes  $o_n$ . The parameters enable researchers to simulate networks that resemble real-world scenarios and to analyze community detection methods under various conditions. LFR networks can serve as valuable resources for testing and comparing algorithm performance.

**Table 1:** Characters of real-world networks

No.	Real network	Nodes	Links
1	Karate	34	78
2	Dolphin	62	159
3	Cornell	183	277
4	Texas	183	279
5	Polbooks	105	441
6	Football	115	613
7	Jazz	198	5484
8	Wisconsin	251	450
9	Facebook3437	534	9626
10	Polblogs	1490	19,025
11	Netscience	1589	2742
12	Yeast	2361	6646

**Table 2:** Parameters used to generate LFR networks

Parameter	Description	Default value
$N$	Number of nodes	NA
$k_{max}$	Maximum degree of nodes	50
$\mu$	Mixing parameter	NA
$t_1$	Negative exponent for the degree sequence	2
$t_2$	Negative exponent for the community size	1
$minc$	Minimum community size	20
$maxc$	Maximum community size	50
$On$	Number of overlapping nodes	NA
$Om$	Number of memberships	2

#### 4.2 Compared Algorithms and Performance Evaluation Indices

To fully demonstrate the advantages and comprehensiveness of the proposed algorithm, MOEA-based methods, graph embedding-based methods, and traditional methods are taken in comparison. In the realm of graph embedding techniques, several benchmark algorithms are considered for comparison, including LINE [32], Node2vec [31], DeepWalk [30], and Word2vec [49]. The mentioned graph embedding-based methods all require pre-configuration of the number of communities. Subsequently, the speaker-listener label propagation algorithm (SLPA) [50] is used as a traditional method of detecting communities by simulating the propagation of information among nodes. For MOEA-based community detection algorithms,

MCMOEAs [18] and MEAs<sub>SN</sub> [51], leverage the MOEA approach to simultaneously optimize multiple objectives. Furthermore, NCMOEAs integrate concepts from both MOEA and graph embedding [52]. In the NCMOEAs framework, the nodes in a network are categorized into two types: candidate central (CC) nodes, which are more likely to serve as community centers, and noncentral (NC) nodes, which are less likely to be central. To effectively represent these different node types, NCMOEAs employ a mixed representation scheme.

In terms of experimental settings, for the traditional and graph embedding-based methods, the parameter configurations are recommended in their respective articles. For MOEA-based methods, a common parameter setup is applied, with a population size ( $PS$ ) of 100 and a maximum generation count ( $gen_{max}$ ) of 50.

Extended modularity  $Q$  and  $NMI$  have been selected as alternative evaluation metrics. The main reason for using Extended Modularity  $Q$  as an evaluation metric in the community detection problem is its ability to accurately measure the quality and cohesion of community structures [53].  $Q$  is computed on the basis of the node assignments in the network and evaluates the degree of separation within communities as follows:

$$Q = \frac{1}{2M} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2M} \right) \delta(C_i, C_j) \quad (8)$$

where  $A_{ij}$  represents the actual number of edges between nodes  $i$  and  $j$ ,  $k_i$  and  $k_j$  are the degrees of nodes  $i$  and  $j$ , respectively,  $M$  represents the total number of edges in the network, and  $\delta(C_i, C_j)$  is an indicator function that takes the value of 1 when nodes  $i$  and  $j$  belong to the same community and a value of 0 otherwise. The advantage of extended modularity  $Q$  lies in its ability to effectively capture connectivity patterns between nodes within and between communities.

The reason for using  $NMI$  as an evaluation metric for community detection is that it serves as a measure of the similarity between the clustering results and the true community structure [54].  $NMI$  is calculated as follows:

$$NMI = \frac{-2 \sum_{i=1}^{C_{true}} \sum_{j=1}^C |C_{ij}| \log(|C_{ij}|N / (|C_i \cap C_j|))}{\sum_{i=1}^{C_{true}} |C_i| \log(|C_i|/N) + \sum_{j=1}^C |C_j| \log(|C_j|/N)} \quad (9)$$

where  $C$  represents the community results,  $C_{true}$  represents the true community structure, and  $C_{ij}$  is the intersection between  $C_i$  and  $C_j$ . Higher values of  $NMI$  indicate a greater similarity between the clustering results and the true community structure, suggesting that the clustering algorithm has accurately captured the underlying community structure. However, Eq. (9) overlooks the importance of the number of communities. In community detection research, the number of generated partitions should be consistent with the true partition in terms of the number of communities. Therefore, Eq. (9) is used as an alternative evaluation metric in place of  $NMI$  to incorporate consideration for the number of communities [55].  $fNMI$  is defined as follows:

$$fNMI = e^{-\frac{|C_{true}-C|}{C_{true}}} \times NMI \quad (10)$$

### 4.3 Experiment Results on Real-World Networks

For the 12 real-world networks, NCMOEAs, MCMOEAs, MEAs<sub>SN</sub>, LINE, Node2vec, DeepWalk, BIGCLAM [56], COPRA, and CPM are compared to SFCMOEA in terms of  $Q$  and *Runtime*.

### 4.3.1 Experimental Results in Terms of Q

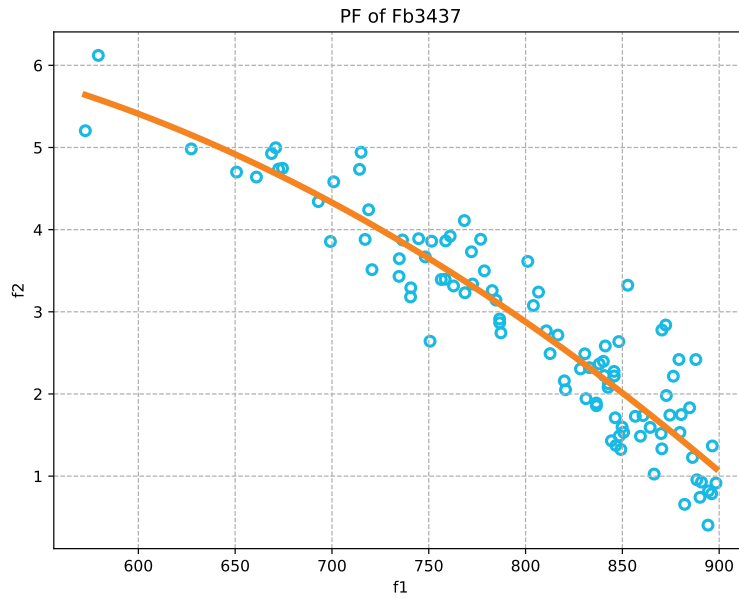
In the experimental results shown in Table 3, SFCMOEA algorithm achieves excellent results in terms of its Q values. For nine out of 12 networks, SFCMOEA outperforms other algorithms in terms of maximum and average values of Q ( $Q_{max}$ ,  $Q_{avg}$ ), including Dolphine, Football, Jazz, Karate, Polblogs, Polbooks, Texas, Cornell, and Wisconsin networks. The statistical results indicate good performance on both large- and small-scale community detection tasks. Taking the Dolphins network as an example, SFCMOEA achieves the highest value of 0.7287 for both  $Q_{max}$  and  $Q_{avg}$ . Performance on Polblogs demonstrates the adaptability of SFCMOEA to dense networks. However, in the Facebook3437 network, SFCMOEA exhibits a lower value of Q compared to other algorithms such as LINE, SLPA, and COPRA, indicating that SFCMOEA may not be well suited for networks with few nodes and excessive overlapping phenomena. It should also be noted that the NCMOEa, which is another MOEA-based algorithm, achieves the best results on the remaining two datasets. In total, MOEA-based algorithms outperform traditional graph embedding-based methods in almost all datasets.

**Table 3:** Comparison on the Q values of SFCMOEA with other algorithms, including MOEA-based, graph embedding-based, and traditional methods (The best results in bold)

Network	Q											
	Measures	NCMOEA	MCMOEa	MEAs_SN	LINE	Node2vec	DeepWalk	SLPA	BIGCLAM	COPRA	CPM	SFCMOEA
Dolphins	Q_max	0.5268	0.5635	0.1737	0.0244	0.4460	0.3550	0.4881	0.0625	0.4592	0.3229	<b>0.7287</b>
	Q_avg	0.5224	0.4376	0.1737	0.0221	0.3787	0.3550	0.3867	0.0358	0.3809	0.3229	<b>0.7287</b>
	Runtime_avg	4.4842	157.0432	4.1615	1.1870	3.0214	0.1499	0.0627	0.3928	0.0234	0.0023	31.5067
Football	Q_max	0.6046	0.5741	0.3185	0.3811	0.6008	0.0470	0.5959	0.4989	0.5931	0.3950	<b>0.7187</b>
	Q_avg	0.5726	0.5378	0.3185	0.2931	0.5928	0.0470	0.5683	0.4581	0.5224	0.3950	<b>0.6904</b>
	Runtime_avg	6.3248	168.3201	10.1604	1.2176	8.7748	0.0483	0.1227	1.2871	0.0901	0.0258	64.8000
Jazz	Q_max	0.4440	0.1918	0.1408	0.1943	0.3625	0.0100	0.4417	0.0161	0.1474	0.0135	<b>0.6335</b>
	Q_avg	0.4439	0.1268	0.1408	0.1752	0.3505	0.0100	0.3767	0.0068	0.0483	0.0135	0.5044
	Runtime_avg	14.4278	181.8446	84.4050	3.6791	26.5231	0.0711	0.5109	4.4835	0.5572	0.4232	87.0000
Karate	Q_max	0.4198	0.6284	0.1641	0.0020	0.2855	0.4190	0.3718	0.2698	0.3594	0.1858	<b>0.7009</b>
	Q_avg	0.4198	0.5487	0.1641	0.0020	0.2725	0.4190	0.1450	0.1650	0.2391	0.1858	<b>0.7009</b>
	Runtime_avg	3.5346	180.8508	0.1295	0.7665	1.6988	0.0330	0.0178	0.1646	0.0115	0.0011	16.7600
Netscience	Q_max	<b>0.9342</b>	0.8853	0.5076	0.0951	0.8360	0.1261	0.8717	0.1212	0.9169	0.7457	0.8507
	Q_avg	<b>0.9205</b>	0.8638	0.5076	0.0625	0.8127	0.1261	0.8474	0.1053	0.9006	0.7457	0.8313
	Runtime_avg	106.0786	231.3652	210.6395	5.0743	149.7497	0.2550	0.0054	132.1328	0.4194	0.1139	109.2133
Polblogs	Q_max	0.3972	0.2884	0.1015	0.1629	0.3672	0.1140	0.4260	0.0149	0.0292	0.0180	<b>0.6582</b>
	Q_avg	0.3970	0.1849	0.1015	0.1536	0.3572	0.1140	0.4044	0.0074	0.0271	0.0180	<b>0.6402</b>
	Runtime_avg	123.9437	579.4012	478.6680	21.7998	206.6383	0.2551	3.6485	119.2856	5.2285	1904.2506	286.1200
Yeast	Q_max	0.5528	0.2916	0.2532	0.0827	<b>0.5597</b>	0.1200	0.5156	0.0252	0.4632	0.2115	0.4839
	Q_avg	0.3023	0.2667	0.2512	0.0562	<b>0.5089</b>	0.1200	0.5012	0.0154	0.3826	0.2115	0.4770
	Runtime_avg	386.1454	696.1657	565.6829	12.5816	324.7257	0.5293	1.8612	306.5399	1.7163	1.2105	248.0133
Facebook 3437	Q_max	<b>0.6792</b>	0.1672	0.1384	0.5157	0.3923	0.0810	0.4790	0.0166	0.4330	0.3319	0.3915
	Q_avg	0.4808	0.1505	0.1287	<b>0.5012</b>	0.3571	0.0810	0.4512	0.0067	0.3612	0.3319	0.3444
	Runtime_avg	42.1497	651.0625	55.1954	5.7817	62.7842	0.1275	1.0538	25.9865	0.9624	5.7256	139.6800
Polbooks	Q_max	0.5269	0.5978	0.2560	0.1915	0.4581	0.1180	0.6710	0.5186	0.4865	0.3644	<b>0.8340</b>
	Q_avg	0.5269	0.4397	0.2560	0.1537	0.4283	0.1180	0.6436	0.4894	0.4253	0.3644	<b>0.7861</b>
	Runtime_avg	5.9773	518.4065	13.1937	1.0218	6.0843	0.0465	0.1016	1.0022	0.0654	0.0241	47.8000
Texas	Q_max	0.5405	0.3382	0.0608	0.1437	0.5182	0.0705	0.2720	0.0890	0.1100	0.1987	<b>0.7051</b>
	Q_avg	0.5213	0.2761	0.0608	0.1076	0.4979	0.0705	0.2304	0.0446	0.0929	0.1987	<b>0.6997</b>
	Runtime_avg	8.219	62.1773	3.4029	0.9052	5.4658	0.6343	0.0416	2.0756	0.0622	0.0034	48.9445
Cornell	Q_max	0.6294	0.5768	0.0734	0.1226	0.6126	0.0412	0.4467	0.0688	0.5694	0.2407	<b>0.7128</b>
	Q_avg	0.6247	0.5216	0.0734	0.0679	0.5991	0.0412	0.3578	0.0259	0.5028	0.2407	<b>0.6623</b>
	Runtime_avg	8.711	56.4513	3.2020	0.8962	5.4315	0.6436	0.0388	2.1642	0.0550	0.0030	48.3610
Wisconsin	Q_max	0.6151	0.5017	0.0655	0.1232	0.5987	0.0186	0.4875	0.0492	0.5043	0.2213	<b>0.7762</b>
	Q_avg	0.6039	0.4428	0.0655	0.0854	0.5829	0.0186	0.4607	0.0196	0.3566	0.2213	<b>0.7627</b>
	Runtime_avg	12.11	62.1773	6.4031	0.9974	8.5422	0.6437	0.0565	3.8175	0.0940	0.0078	51.2687



Furthermore, the performance of the SFCMOEA algorithm can be further evaluated by examining the Pareto front. For MOEA-based community detection algorithms, a good quality of the generated communities corresponds to the objectives of maximizing  $f_1$  and minimizing  $f_2$ . Fig. 7 shows the Pareto front [57] obtained by SFCMOEA on the Fb3437 dataset. In Fig. 7, there are 50 individuals, with each data point representing the  $f_1$  and  $f_2$  values of a different individual. The upper point approaches the optimal value for  $f_2$ , while the right upper point satisfies  $f_1$  to the greatest extent. The points form a curve, representing the Pareto front, that is, the set of non-dominated solutions in the context of the MOP. Moreover, the curve indicates that the proposed algorithm is capable of finding a good balance between the two objectives.



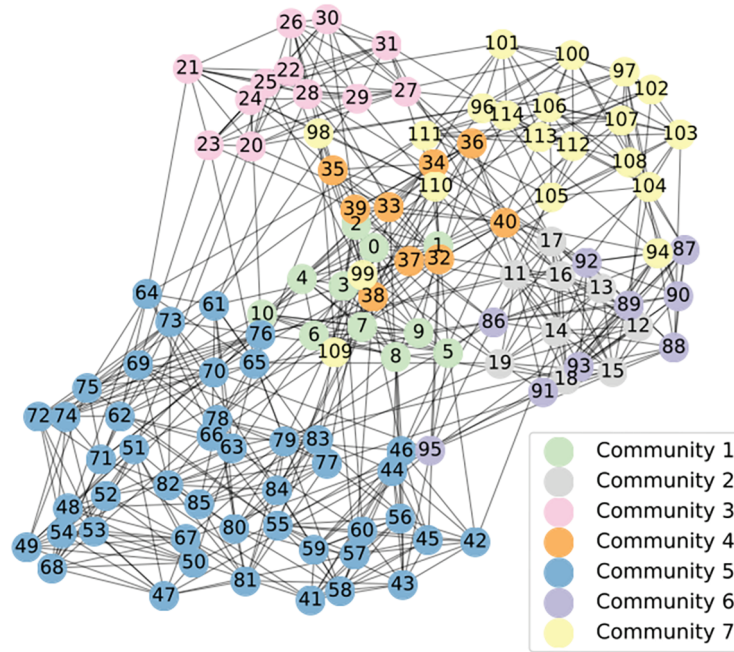
**Figure 7:** Illustration of the mapping of the Pareto Front ( $f_1 = \text{KKM}$ ,  $f_2 = \text{RC}$ )

Furthermore, Fig. 8 presents the results obtained by running SFCMOEA on the Football network. From the figure, it can be observed that 7 communities are discovered in the network, differentiated by different colors. Each partitioned result exhibits strong internal connections while the external connections remain sparse, consistent with the objectives of community detection tasks.

#### 4.3.2 Experimental Results in Terms of Runtime

In terms of runtime, SFCMOEA is at a moderate level in most networks in terms of average runtime  $\text{Runtime}_{avg}$ . Compared to MOEA-based methods, SFCMOEA performs better than MCMOEa and MEAs\_SN at all data sizes and can perform equally to or better than the NCMOEa algorithm in larger-scale scenarios. However, for graph embedding-based algorithms such as Node2vec and DeepWalk, SFCMOEA has a noticeably longer runtime. Among traditional algorithms, SLPA demonstrates relatively short runtimes compared to the graph embedding-based methods in most networks. In particular, on data sets such as Fb3437 and Polblogs, in which the number of edges is much higher than the number of nodes, SFCMOEA achieves a  $\text{Runtime}_{avg}$  close to that of Node2vec, indicating that SFCMOEA has the potential to address efficiency issues in dense networks. The comparison between SFCMOEA and its ablated counterpart, SFCMOEA-NF, reveals a significant difference in  $\text{Runtime}_{avg}$ . The ablation experiment demonstrates that

the SFNE process enhances the computational efficiency of the algorithm while simultaneously enhancing its partition quality.

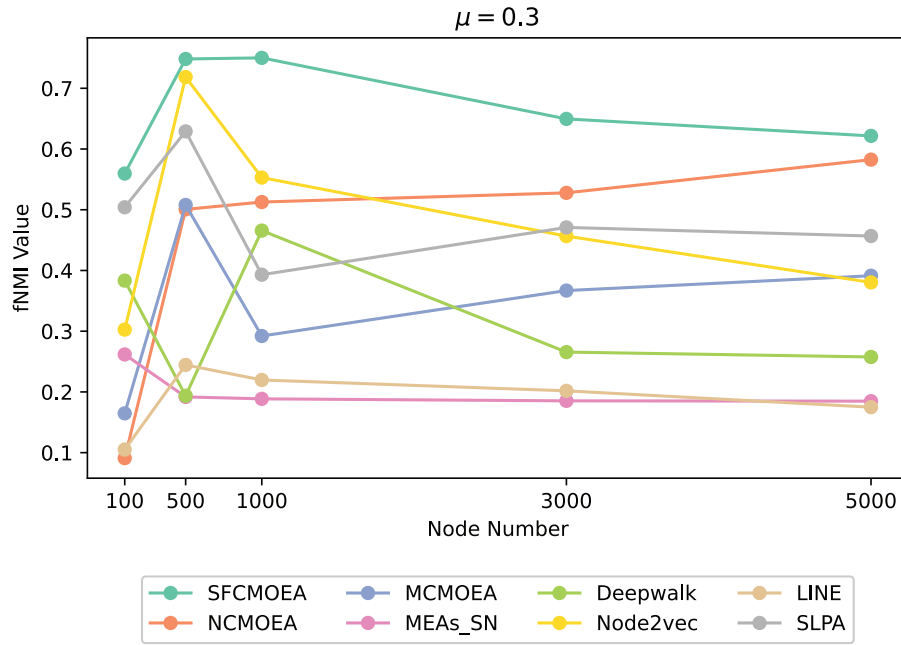


**Figure 8:** Community detection results obtained by applying SFCMOEA to the Football network where different colors represent different communities

#### 4.4 Experiment Results on Synthetic Networks

The parameters used to construct the LFR networks for these experiments are presented in Table 2. There are a total of nine distinct parameters, of which six parameters have been set to their default values as listed in the table. The remaining parameters  $N$ ,  $\mu$ , and  $O_n$  have been configured based on the specific network scales needed for the experiments. Specifically, a series of LFR datasets is generated by setting the parameters to  $N = \{100, 500, 1000, 3000, 5000\}$ ,  $\mu = \{0.3\}$ , and  $O_n = 1/10N$ . Eight different algorithms are compared using LFR datasets, namely SFCMOEA, NCMOEA, MCMOEA, MEAs\_SN, LINE, Node2vec, DeepWalk and SLPA. Since LFR algorithms generate ground truth partitions, the main comparison metric used is the value of  $fNMI$ , which provides a comprehensive evaluation that considers the number of partitions.

In Fig. 9, compared to the other algorithms, SFCMOEA performs better on datasets characterized by a high degree of community overlap and a larger scale. As the network becomes more complex, other algorithms show a significant decrease in performance, whereas the performance of SFCMOEA remains stable. In the case of sparse networks, both the SLPA and Node2vec algorithms are close to the SFCMOEA. From the perspective of community partitioning, SLPA and Node2vec algorithms generate more number of community structures. Interestingly, as the network scale increases, the performance of the NCMOEA algorithm actually improves. In general, in terms of evaluating the results of network construction, graph embedding-based methods may exhibit some limitations.



**Figure 9:** Example results of eight algorithms based on the  $fNMI$  evaluation metric

In addition, the runtimes on the constructed LFR network datasets, the evaluation criterion of runtime, is also taken into account. A comparison is made among five algorithms using five different scales of LFR datasets. The algorithms compared included traditional algorithms, graph embedding algorithms, and MOEA-based algorithms simultaneously. Table 4 presents the  $Runtime_{avg}$  results of five algorithms on datasets of different sizes. When comparing SFCMOEA to other algorithms, several advantages and disadvantages can be observed. One notable advantage of SFCMOEA is its relatively shorter runtime compared to algorithms such as NCMOE and MCMOE. In smaller datasets with 100 and 500 nodes, SFCMOEA demonstrates a significant reduction in runtime, taking only 44 and 73 s, respectively. However, as the size of the data set increases to 1000, 3000, and 5000 nodes, the runtime of SFCMOEA also increases. The observed increase in runtime suggests that SFCMOEA may face scalability issues and experience longer execution times when dealing with larger networks. At the same time, MOEA-based methods fall significantly behind LINE and SLPA in terms of runtime, while SFCMOEA maintains results that are close to the graph embedding method Node2vec. In general, compared to MCMOE, SFCMOEA achieves improvements in both the quality of the resulting partitions and the running time by employing the SFNE method to obtain an advantageous starting vertex set for the search process. In contract, as seen from Table 4, MOEA-based methods require more time, still challenging to compete with traditional methods that do not involve time-consuming MOEA stages but require the prior setting of the number of communities.

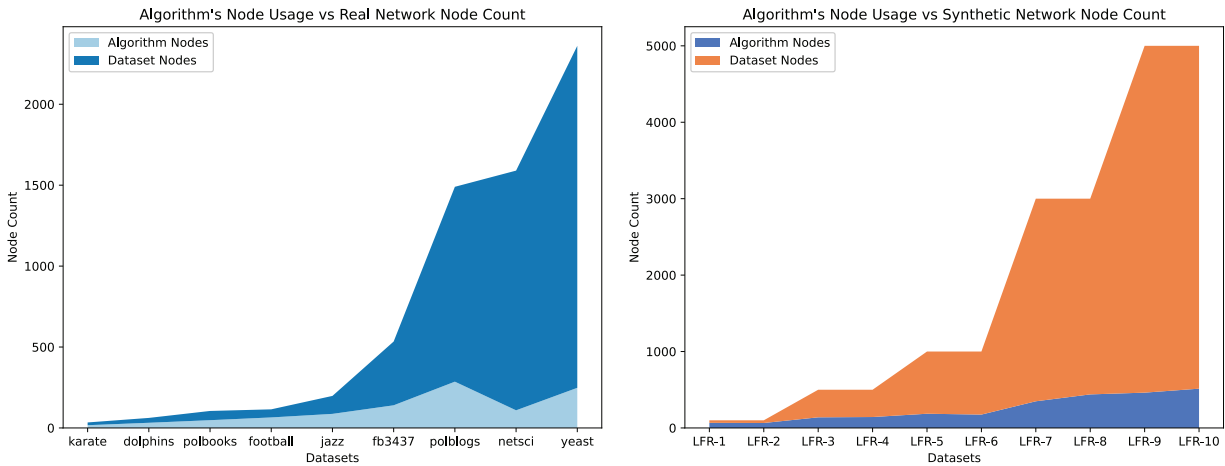
#### 4.5 Performance Analysis on SFNE Method

This section aims to highlight the effectiveness of the proposed SFNE method by investigating its behavior on both real-world and synthetic network datasets. The number of nodes in the network and the size of the structural feature vertex set used by SFCMOEA is set as the comparison standard. MOEA-based methods, graph embedding methods and traditional methods typically perform the search starting from all nodes as the compared algorithms. In this evaluation, LFR benchmarks generated with parameter settings of  $\mu = 0.1$  and  $\mu = 0.3$  are used.

**Table 4:** Comparison on the average runtimes among five algorithms under various scales of LFR networks

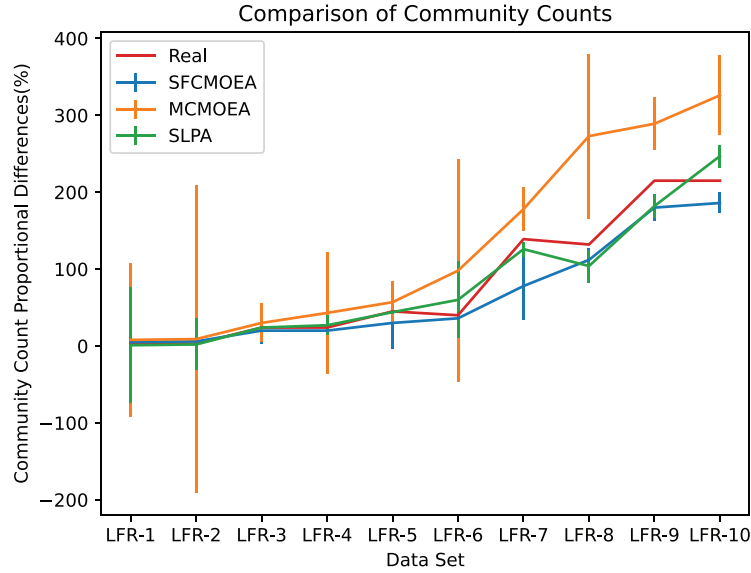
N	Average runtime (s)				
	NCMOEA	MCMOE	SLPA	Node2vec	SFCMOEA
100	7	263	1	6	44
500	42	322	1	57	73
1000	98	348	4	122	123
3000	729	1639	12	446	608
5000	2112	4928	14	876	1496

From the two figures shown in Fig. 10, as the scale of the network increases, the ratio of the number of nodes utilized in SFCMOEA to all nodes in the network continues to expand. Even when the network scale is small, there can be almost 50% savings in node usage. Combined with the evaluations of  $Q$ ,  $fNMI$ , and  $Runtime_{avg}$ , it indicates that the set of structural features that generate the SFNE method can effectively represent communities and the underlying network structures. While the network scale increases, the efficiency of node utilization also increases, which represents that the SFCMOEA has adaptability to datasets of different scales. The improvement in node utilization also suggests that the algorithm consistently maintains a high level of utilization of computer resources.

**Figure 10:** Comparison on the number of nodes actually used by the SFCMOEA algorithm to the total number of nodes in real and synthetic networks, respectively. The upper line represents the total number of network nodes, while the lower line indicates the number of nodes used in the search process

For further analysis, Fig. 11 presents a comparison of the number of detected partitions with the actual number of communities in 10 different scaled LFR datasets generated by SFCMOEA, MCMOE and SLPA algorithms. The MOEA-based methods and traditional methods are chosen for comparison to provide a comprehensive analysis of the results. From the figure, SFCMOEA is closest to the actual number of communities in almost all networks, with consistent results in the cases of LFR4 and LFR6. The reason can be attributed to the vertex set generated by the SFNE step, which ensures that the number of communities generated in the subsequent community search step is close to the actual partition count. As the network scale increases, the MCMOE algorithm tends to generate a larger number of communities, which is also reflected in the decreasing  $fNMI$  results. In contrast, the SLPA method performs relatively better than the

MOEA-based methods in this regard. Considering Fig. 11 in conjunction with Fig. 10, SFCMOEA achieves a balance by minimizing the use of nodes during the search while ensuring the reliability of the community count generated.



**Figure 11:** Comparison between the number of communities detected by three algorithms (SFCMOEA, MCMOE, SLPA) and the actual number of communities. Each data point represents the average number of communities across different datasets, with error bars indicating the maximum and minimum values

#### 4.6 Experiment Results on Large-Scale Networks

In order to validate the feasibility and efficiency of SFCMOEA on large-scale real-world datasets, the ca-Rr and ca-HePt datasets are selected. These two datasets consist of 5242 and 9877 nodes, respectively, that make up the network. In Table 5, SFCMOEA performs well in the community detection task in large-scale networks, while addressing the limitations of MCMOE. Furthermore, compared to NCMOE, an MOEA-based algorithm that performs well on small-scale networks, the SFCMOEA algorithm demonstrates advantages in larger-scale scenarios. However, there is still some gap between SFCMOEA and non-MOEA-based methods.

**Table 5:** Comparison on the runtimes in large scale real networks

Network	Runtimes (s)				
	NCMOEA	MCMOE	SLPA	Node2vec	SFCMOEA
ca-Gr	1557	NA	3	44	818
ca-HepT	6382	NA	7	73	1327

## 5 Conclusions

In this paper, a novel algorithm is proposed for community detection in overlapping networks, called SFCMOEA. The SFCMOEA algorithm addresses the limitations of existing community detection algorithms. First, to reduce the search space and speed up the search for optimal solutions, the SFNE method is designed to generate a set of vertices that collect the most influential nodes for community detection.

Only the nodes within the vertex set are involved in subsequent search steps. To identify overlapping nodes, a maximal clique representation scheme is employed on the generated vertex set. A link strength-based method is then proposed to measure the interconnections between cliques, thereby improving the quality of the detected communities. Finally, the HCMOEA approach is proposed to address cliques and the remaining unexplored nodes separately, based on the clique-structured individuals. The HCMOEA approach enables the generation of a number of communities closer to the actual partition count, followed by the allocation of nodes according to the attractiveness of different communities. The number of individuals is effectively reduced while enhancing the efficiency of the algorithm. In the experiments, various networks have been used to test SFCMOEA against some well-known algorithms, including non-EA-based, MOEA-based, and graph embedding-based methods for community detection. In terms of comparing the generated number of communities, SFCMOEA shows a closer number to real partitions and greater stability. The main drawback of SFCMOEA lies in the runtime compared to the traditional methods. Therefore, in the future, algorithm parallelization based on CPU or GPU is required. Furthermore, we plan to improve SFCMOEA by incorporating additional features, such as node attributes and edge weights, and the performance of SFCMOEA on dynamic networks will also be considered.

**Acknowledgement:** None.

**Funding Statement:** This research was supported in part by the National Natural Science Foundation of China under Grants 62473176, 62073155, 62002137, 62106088, and 62206113, in part by National Key Laboratory of Ship Structural Safety under Grant 450324300, in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX24\_2642.

**Author Contributions:** The authors confirm contribution to the paper as follows: Methodology, Sicheng Ma; writing—original draft preparation, Lixiang Zhang; validation, Guocai Chen and Zeyu Dai; writing—review and editing, Junru Zhu; supervision, funding acquisition, Wei Fang. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** There are no new data or materials for this review.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Zhuo Z, Chen B, Yu S. Overlapping community detection via adaptive Bayesian nonnegative matrix tri-factorization. *IEEE Trans Netw Sci Eng.* 2026;13:2951–63. doi:10.1109/tnse.2025.3623790.
2. Girvan M, Newman ME. Community structure in social and biological networks. *Proc Nat Acad Sci.* 2002;99(12):7821–6. doi:10.1073/pnas.122653799.
3. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *J Stat Mech Theory Exp.* 2008;2008(10):P10008. doi:10.1088/1742-5468/2008/10/p10008.
4. Rosvall M, Bergstrom CT. Maps of random walks on complex networks reveal community structure. *Proc Nat Acad Sci.* 2008;105(4):1118–23. doi:10.1073/pnas.0706851105.
5. Malliaros FD, Vazirgiannis M. Clustering and community detection in directed networks: a survey. *Phys Rep.* 2013;533(4):95–142.
6. Burgess M, Adar E, Cafarella M. Link-prediction enhanced consensus clustering for complex networks. *PLoS One.* 2016;11(5):e0153384. doi:10.1371/journal.pone.0153384.
7. Xie H, Ying X, Wang X, Qi Y, Chen W, Huang X, et al. Redundancy-aware masked graph autoencoder for overlapping community detection in attributed networks. *Eng Appl Artif Intell.* 2025;162:112821. doi:10.1016/j.engappai.2025.112821.



8. Zhang L, Pan H, Su Y, Zhang X, Niu Y. A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection. *IEEE Trans Cybern.* 2017;47(9):2703–16. doi:10.1109/tcyb.2017.2711038.
9. Adamic L, Adar E. How to search a social network. *Soc Netw.* 2005;27(3):187–203. doi:10.1016/j.socnet.2005.01.007.
10. Souravlas S, Anastasiadou SD, Economides T, Katsavounis S. Probabilistic community detection in social networks. *IEEE Access.* 2023;11:25629–41. doi:10.1109/access.2023.3257021.
11. Arnaudon A, Schindler DJ, Peach RL, Gosztolai A, Hodges M, Schaub MT, et al. PyGenStability: multiscale community detection with generalized Markov stability. *arXiv:2303.05385.* 2023.
12. Ali M, Hassan M, Kifayat K, Kim JY, Hakak S, Khan MK. Social media content classification and community detection using deep learning and graph analytics. *Technol Forecast Soc Change.* 2023;188:122252. doi:10.1016/j.techfore.2022.122252.
13. Pizzuti C. A multi-objective genetic algorithm for community detection in networks. In: 2009 21st IEEE international conference on tools with artificial intelligence. Piscataway, NJ, USA: IEEE; 2009. p. 379–86.
14. Zhang W, Shang R, Jiao L. Complex network graph embedding method based on shortest path and MOEA/D for community detection. *Appl Soft Comput.* 2020;97:106764. doi:10.1016/j.asoc.2020.106764.
15. Shang R, Wang S, Zhang W, Feng J, Jiao L, Stolkin R. Evolutionary multi-objective overlapping community detection based on fusion of internal and external connectivity and correction of node intimacy. *Appl Soft Comput.* 2024;154:111414. doi:10.2139/ssrn.4288728.
16. Shang R, Zhao K, Zhang W, Feng J, Li Y, Jiao L. Evolutionary multiobjective overlapping community detection based on similarity matrix and node correction. *Appl Soft Comput.* 2022;127:109397. doi:10.1016/j.asoc.2022.109397.
17. Gong M, Ma L, Zhang Q, Jiao L. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Phys Stat Mech Appl.* 2012;391(15):4050–60. doi:10.1016/j.physa.2012.03.021.
18. Wen X, Chen WN, Lin Y, Gu T, Zhang H, Li Y, et al. A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. *IEEE Trans Evol Comput.* 2016;21(3):363–77. doi:10.1109/tevc.2016.2605501.
19. Pizzuti C. Evolutionary computation for community detection in networks: a review. *IEEE Trans Evol Comput.* 2017;22(3):464–83. doi:10.1109/tevc.2017.2737600.
20. Su X, Xue S, Liu F, Wu J, Yang J, Zhou C, et al. A comprehensive survey on community detection with deep learning. *IEEE Trans Neural Netw Learn Syst.* 2024;35(4):4682–702. doi:10.1109/tnnls.2021.3137396.
21. De Santo A, Galli A, Moscato V, Sperli G. A deep learning approach for semi-supervised community detection in Online Social Networks. *Knowl Based Syst.* 2021;229:107345. doi:10.1016/j.knosys.2021.107345.
22. Liu H, Wei J, Xu T. Community detection based on community perspective and graph convolutional network. *Expert Syst Appl.* 2023;231:120748. doi:10.1016/j.eswa.2023.120748.
23. Sismanis K, Potikas P, Souliou D, Pagourtzis A. Overlapping community detection using graph attention networks. *Future Gener Comput Syst.* 2025;163:107529. doi:10.1016/j.future.2024.107529.
24. Chen J, Gong Z, Mo J, Wang W, Wang W, Wang C, et al. Self-training enhanced: network embedding and overlapping community detection with adversarial learning. *IEEE Trans Neural Netw Learn Syst.* 2022;33(11):6737–48. doi:10.1109/tnnls.2021.3083318.
25. Zhang W, Wang W, Shang R, Xu S. Overlapping community detection based on graph attention autoencoder and self-trained clustering. *Appl Soft Comput.* 2025;183:113584. doi:10.1016/j.asoc.2025.113584.
26. Xie Y, Gong M, Wang S, Yu B. Community discovery in networks with deep sparse filtering. *Pattern Recognit.* 2018;81:50–9. doi:10.1016/j.patcog.2018.03.026.
27. Cheng J, He C, Lin X, Liu W, Han K, Tang Y. When graph neural networks meet deep nonnegative matrix factorization: an encoder and decoder-like method for community detection. *Expert Syst Appl.* 2025;271:126676. doi:10.1016/j.eswa.2025.126676.
28. Bi XA, Chen K, Jiang S, Luo S, Zhou W, Xing Z, et al. Community graph convolution neural network for Alzheimer's Disease classification and pathogenetic factors identification. *IEEE Trans Neural Netw Learn Syst.* 2025;36(2):1959–73. doi:10.1109/tnnls.2023.3269446.
29. Škrlj B, Kralj J, Lavrač N. Embedding-based Silhouette community detection. *Mach Learn.* 2020;109:2161–93. doi:10.1007/s10994-020-05882-8.

30. Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. New York, NY, USA: ACM; 2014. p. 701–10.
31. Grover A, Leskovec J. node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. New York, NY, USA: ACM; 2016. p. 855–64.
32. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. LINE: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web; 2015 May 18–22. Florence, Italy. p. 1067–77.
33. Xu R, Che Y, Wang X, Hu J, Xie Y. Stacked autoencoder-based community detection method via an ensemble clustering framework. *Inf Sci.* 2020;526:151–65. doi:10.1016/j.ins.2020.03.090.
34. Zhao Y, Li W, Liu F, Wang J, Luvembe AM. Integrating heterogeneous structures and community semantics for unsupervised community detection in heterogeneous networks. *Expert Syst Appl.* 2024;238:121821. doi:10.1016/j.eswa.2023.121821.
35. Hamilton WL, Ying R, Leskovec J. Representation learning on graphs: methods and applications. arXiv:1709.05584. 2017.
36. Xu H, Zeng W, Zeng X, Yen GG. An evolutionary algorithm based on Minkowski distance for many-objective optimization. *IEEE Trans Cybern.* 2018;49(11):3968–79. doi:10.1109/tcyb.2018.2856208.
37. Xie J, Kelley S, Szymanski BK. Overlapping community detection in networks: the state-of-the-art and comparative study. *ACM Comput Surv.* 2013;45(4):1–35. doi:10.1145/2501654.2501657.
38. Palla G, Derényi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature.* 2005;435(7043):814–8. doi:10.1038/nature03607.
39. Ma W, Cai L, He T, Chen L, Cao Z, Li R. Local expansion and optimization for higher-order graph clustering. *IEEE Internet Things J.* 2019;6(5):8702–13. doi:10.1109/jiot.2019.2923228.
40. El Kouni IB, Karoui W, Romdhane LB. Node importance based label propagation algorithm for overlapping community detection in networks. *Expert Syst Appl.* 2020;162:113020. doi:10.1016/j.eswa.2019.113020.
41. Zhu W, Li H, Wei W. A two-stage multi-objective evolutionary algorithm for community detection in complex networks. *Mathematics.* 2023;11(12):2702. doi:10.3390/math11122702.
42. Zhang L, Yang H, Yang S, Zhang X. A macro-micro population-based co-evolutionary multi-objective algorithm for community detection in complex networks [Research Frontier]. *IEEE Comput Intell Mag.* 2023;18(3):69–86. doi:10.1109/mci.2023.3277773/mml.
43. Rahimi S, Abdollahpouri A, Moradi P. A multi-objective particle swarm optimization algorithm for community detection in complex networks. *Swarm Evol Comput.* 2018;39:297–309. doi:10.1016/j.swevo.2017.10.009.
44. Ribeiro LF, Saverese PH, Figueiredo DR. struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. New York, NY, USA: ACM; 2017. p. 385–94.
45. Dhillon IS, Guan Y, Kulis B. Kernel k-means: spectral clustering and normalized cuts. In: Proceedings of the Tenth ACM SIGKDD international conference on knowledge discovery and data mining. New York, NY, USA: ACM; 2004. p. 551–6.
46. Hagen L, Kahng AB. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans Comput-Aided Design Integr Circ Syst.* 1992;11(9):1074–85. doi:10.1109/43.159993.
47. Lancichinetti A, Fortunato S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys Rev E.* 2009;80(1):016118. doi:10.1103/physreve.80.016118.
48. Kamiński B, Prałat P, Théberge F. Artificial benchmark for community detection (abcd)—fast random graph model with community structure. *Netw Sci.* 2021;9(2):153–78. doi:10.1017/nws.2020.45.
49. Goldberg Y, Levy O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv:1402.3722. 2014.
50. Xie J, Szymanski BK, Liu X. Slpa: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In: 2011 IEEE 11th international conference on data mining workshops. Piscataway, NJ, USA: IEEE; 2011. p. 344–9.

51. Liu C, Liu J, Jiang Z. A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. *IEEE Trans Cybern.* 2014;44(12):2274–87. doi:10.1109/tcyb.2014.2305974.
52. Yang H, Li B, Cheng F, Zhou P, Cao R, Zhang L. A node classification-based multiobjective evolutionary algorithm for community detection in complex networks. *IEEE Trans Comput Soc Syst.* 2024;11(1):292–306. doi:10.1109/tcss.2022.3223159.
53. Shen H, Cheng X, Cai K, Hu MB. Detect overlapping and hierarchical community structure in networks. *Phys Stat Mech Appl.* 2009;388(8):1706–12. doi:10.1016/j.physa.2008.12.021.
54. Lancichinetti A, Fortunato S, Kertész J. Detecting the overlapping and hierarchical community structure in complex networks. *New J Phys.* 2009;11(3):033015. doi:10.1088/1367-2630/11/3/033015.
55. Amelio A, Pizzuti C. Is normalized mutual information a fair measure for comparing community detection methods? In: *Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining 2015*. New York, NY, USA: ACM; 2015. p. 1584–5.
56. Yang J, Leskovec J. Overlapping community detection at scale: a nonnegative matrix factorization approach. In: *WSDM' 13*. New York, NY, USA: Association for Computing Machinery; 2013. p. 587–96. doi:10.1145/2433396.2433471.
57. Van Veldhuizen DA, Lamont GB. Evolutionary computation and convergence to a pareto front. In: Koza JR, editor. *Late breaking papers at the genetic programming 1998 conference*. Madison, WI, USA: Stanford University Bookstore; 1998. p. 221–8.