Applied Soft Computing 127 (2022) 109397

Contents lists available at ScienceDirect

Applied Soft Computing

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

Evolutionary multiobjective overlapping community detection based on similarity matrix and node correction

Ronghua Shang, Kejia Zhao, Weitong Zhang *, Jie Feng, Yangyang Li, Licheng Jiao

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi Province 710071, China

ARTICLE INFO

Article history:
Received 2 April 2022
Received in revised form 11 July 2022
Accepted 21 July 2022
Available online 26 July 2022

Keywords:
Fuzzy clustering
Similarity matrix
Evolutionary multiobjective
Overlapping community detection

ABSTRACT

The method of overlapping community detection based on fuzzy clustering is sensitive to the initialization of community centers, which easily traps in local optima and leads to node misclassification. This paper proposes an evolutionary multiobjective algorithm based on similarity matrix and node correction to detect overlapping communities to solve the above problems. Firstly, the algorithm determines a similarity community for each node by setting the similarity threshold. Then, the central nodes are found more accurately through the similarity distribution of the similarity communities. Secondly, under the framework of the evolutionary multiobjective algorithm, the similarity communities of the central nodes are used as the initial communities to obtain the nonoverlapping communities. In addition, the algorithm proposes a correction strategy for the noncentral nodes based on the similarity communities. The correction strategy obtains the adjacent nodes of each node's similarity community. It then uses each adjacent node's community to correct the nonoverlapping community. Finally, the algorithm adjusts the noncentral nodes' correction strategy. This correction strategy corrects the overlapping nodes according to the number of each overlapping node's labels. It takes the separation operation to further correct overlapping nodes to obtain the corrected overlapping communities. This paper uses seventeen real networks and a variety of synthetic networks with different parameters to verify the proposed algorithm's effectiveness. And the proposed algorithm achieves higher accuracy of community detection in most networks than four state-of-the-art overlapping community detection algorithms.

1. Introduction

In many areas of the real world, network science models real systems within them and analyzes those real systems [1], such as social networks [2], scientific collaboration networks [3], biological networks [4], etc. Nodes in the network can represent the entities in the system, and edges connect the entities [5]. The analysis process of these networks is the disclosure of their topological characteristics [6], such as small-world [7] and scale-free [8], which can be studied in depth to increase the understanding of complex networks and thus better mine the potential information contained in data. In recent years, community structure has become one of the critical directions in studying network properties [9] and is highly influential in the development of fields related to data mining [10]. Although related research scholars have proposed many excellent community detection algorithms [11,12], there is no definition of community structure that has been generally accepted by researchers so far [13]. For a complex network with a distinct community structure, the community structure refers to dividing a network into communities, where nodes within the same community have dense connections and different communities have sparse connections [5]. Usually, for the same community in a network, its nodes share essentially the same attributes or have more similar functions. Therefore, along with the rapid development of modern technology and the fast transmission of network information, community structure detection in complex networks has become a necessary technique in network science, social science, and physical science [14].

Many scholars have proposed algorithms for community structure in complex networks, such as Newman's fast algorithm [15], the extreme value optimization algorithm [16] and algorithm based on modularity and simulated annealing method [17]. These algorithms focus on discovering nonoverlapping communities, i.e., there are no duplicate nodes among the communities divided in the network. However, in some real-world networks, the passage of time may lead to gradual changes in different stages of communities. As communities continue to change, there is a tendency to overlap [18]. In overlapping networks, some

* Corresponding author.
E-mail address: wtzhang_1@xidian.edu.cn (W. Zhang).

https://doi.org/10.1016/j.asoc.2022.109397
1568-4946/

communities intersect with each other, i.e., some nodes have multiple community identities [19]. For example, in social networks, people may join various clubs. Due to the diversity of many scientists' research fields, their corresponding identification information is no longer limited to a single. Thus, the practical significance of dividing overlapping communities can be reflected from multiple perspectives of the real world. In general, overlapping communities can be classified into two types: clear and fuzzy [20]. The clear overlapping communities mainly refer to the fact that every node in the network entirely belongs to the community associated with it. In contrast, the fuzzy overlapping communities mainly refer to the fact that some nodes in the network belong to communities with varying attribution factors.

To effectively detect overlapping communities in complex networks, the algorithms of overlapping community detection have been proposed by some scholars, such as Palla et al. [21] used effective technique to explore large-scale overlapping communities, an algorithm based on the concept of splitting [22], Lancichinetti et al. [23] used local optimization to find overlapping communities, and so on. More and more scholars have improved the evolutionary multiobjective algorithm and applied it in other fields. For instance, Zhang et al. [24] proposed determinantal point processes to balance convergence and population diversity in high-dimensional objective space. In time series classification, Gong et al. [25] simultaneously optimized representation capacity, separation capacity, and network size to obtain the advantage of each objective. In neural network ensembles, Chen et al. [26] defined the crossover operator and the mutation operator, followed by finding the best trade-off among empirical error, correlation, and regularization. In software project scheduling, Shen et al. [27] balanced multiple objectives: satisfaction, duration, robustness, and cost. In job shop scheduling, Shen et al. [28] achieved dynamic and flexible job shop scheduling by processing multiple objectives simultaneously and combining multiple heuristics. Many scholars have also achieved good results using evolutionary multiobjective in overlapping community detection algorithms. The fuzzy clustering technique allowed a node to appear in different clustering results, which then achieved the effect of overlap between multiple clusters. Therefore, the fuzzy clustering has become one of the effective overlapping community detection techniques [29]. Over the past decade or so, the methods of overlapping community detection based on the fuzzy clustering have been continuously proposed. For example, Psorakis et al. [30] proposed a probabilistic approach based on the model of Bayesian non-negative matrix factorization for overlapping community detection. The approach obtained the feature matrix by non-negative matrix factorization. Then, the feature matrix was combined into model parameter inference to get the soft assignment. Wu et al. [31] proposed a dynamic clustering method for overlapping community detection via network oscillators. The evolution of nodes with random initial phases was performed by designing a specialized set of differential equations. The phase of the overlapping nodes quantified the affiliation of the communities to which the overlapping nodes belong. Wang et al. [32] implemented fuzzy overlapping community detection by introducing local random wandering-based distances. The method measured the local distance using the similarity index based on random wandering and found overlapping communities by fuzzy c-means clustering.

Although the above fuzzy clustering-based overlapping community detection algorithms can be well used to discover overlapping communities, their effectiveness and detection accuracy still need to be improved. Moreover, the following problems existed in using the fuzzy clustering methods. For example, Wang et al. [33] proposed a fuzzy c-mean clustering algorithm based on particle swarm optimization by analyzing the advantages and disadvantages of the fuzzy c-mean clustering algorithm. Izakian et al. [34] proposed a hybrid fuzzy clustering method using the benefits of fuzzy c-mean clustering algorithm and fuzzy particle swarm algorithm. Wikaisuksakul et al. [35] proposed a multiobjective evolutionary method that did not require the number of clusters. The method introduced an adaptive mechanism to combine the fuzzy c-mean clustering algorithm with NSGA-II. Since the fuzzy clustering was more sensitive in initializing community centers, it led to the fact that the divisions obtained by these methods were prone to local optima. After that, Havens et al. [36] proposed the new formula for fuzzy validity metrics by generalizing the modularity function. Wang et al. [37] introduced a structural similarity based on local interactions between adjacent vertices to measure the fuzzy relationship between vertices. Ding et al. [38] identified and removed all links in the derived link community by the node clustering technique. Then, a network decomposition-based overlapping community detection algorithm was proposed. Biswas et al. [39] proposed a fuzzy cohesive community detection method by iteratively updating node affiliation. However, before these algorithms could be run, it was also necessary to define the number of communities or the relevant parameters for fuzzy clustering. These parameters could hardly be determined in advance, but they played a crucial role in the clustering results.

This paper proposes an evolutionary multiobjective algorithm based on similarity matrix and node correction for overlapping community detection. The proposed algorithm can solve the high sensitivity of the community centers' initialization and correct misclassified nodes. Firstly, the algorithm uses the diffusion kernel similarity to calculate the similarity matrix and sets the similarity threshold based on the average similarity between nodes. Nodes with a similarity greater than the similarity threshold are used as the adjacent nodes in the similarity communities. After that, the adjacent nodes are used to determine the similarity communities of each node. The total similarity of each node's similarity community is calculated and sorted in descending order, and then the central nodes are selected from the sort in turn. At the same time, the adjacent nodes in the central nodes' similarity communities are deleted from the sort. Secondly, under the framework of the evolutionary multiobjective algorithm, the similarity communities of the central nodes are used as the initial communities. Then central nodes in the network are optimized by minimizing the kernel k-means and the ratio cut to obtain nonoverlapping community detection. In addition, because the pre-division may have the wrong partition of noncentral nodes, a similarity community based correction strategy for noncentral nodes is proposed. The adjacent nodes of each node are obtained according to the similarity communities. Then the adjacent nodes with increased modularity are selected to correct the nodes after the nodes are changed to the community to which each adjacent node belongs. Finally, because overlapping nodes appear in dividing overlapping communities, the above correction strategy needs to be adjusted. The algorithm sorts the adjacent nodes' labels of the overlapping nodes in descending order. Then, according to the number of community labels of the overlapping nodes, the algorithm selects the same number of labels from the sort to correct the overlapping nodes. And the overlapping nodes are further corrected by taking the separation operation. After that, the final corrected overlapping communities are obtained. The main contributions of this paper are as follows:

(1) The similarity matrix is calculated by using the diffusion kernel similarity. Then, the similarity threshold is set to divide the similarity community of each node and determine the central nodes more precisely.

(2) The correction strategy of noncentral nodes is proposed based on the similarity community in the pre-division. The nodes'

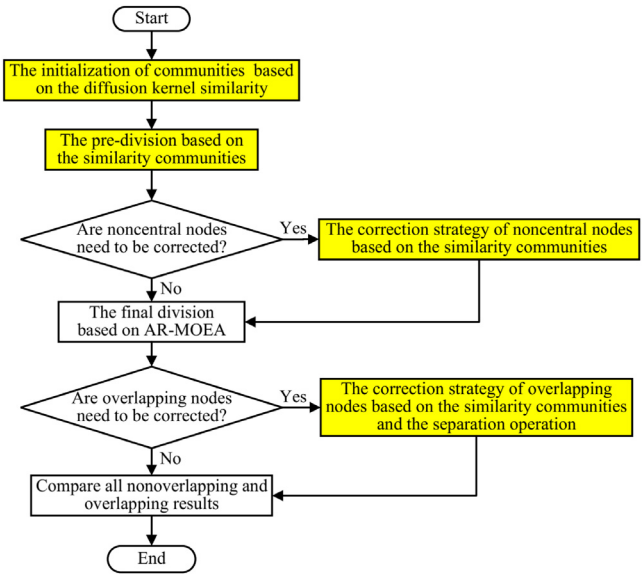

Fig. 1. The working framework of the proposed algorithm.

adjacent nodes are obtained based on the similarity community so that the nonoverlapping communities can be corrected to improve the results of pre-division .

(3) The above correction strategy is adjusted to correct the overlapping nodes according to the partition of adjacent nodes in the similarity community of overlapping nodes. And the separation operation further corrects the overlapping nodes.

The rest of the paper's structure is below. Section 2 describes the proposed algorithm in detail from the perspective of similarity matrix and node correction. Section 3 provides an experimental comparison of real networks and synthetic networks. Section 4 summarizes this paper in combination with the experimental part.

2. Proposed method

To solve the problem of the high sensitivity of community centers' initialization and to judge and correct the nodes with the wrong partition. This paper proposes an evolutionary multiobjective overlapping community detection algorithm based on similarity matrix and node correction, whose working framework is shown in Fig. 1.

The working framework of the proposed algorithm is given in Fig. 1. A four-part approach is proposed in this paper. The community initialization based on the diffusion kernel similarity, the pre-division based on the similarity community, the correction strategy of noncentral nodes based on the similarity community, and the correction strategy of overlapping nodes based on the similarity community and the separation operation. These four parts are described in the following.

2.1. Community initialization based on diffusion kernel similarity

Traditional algorithms used the adjacency matrix to determine the nodes' degree directly, then selected the nodes as the central nodes based on the relationship of nodes' degree. However, the relationship of nodes' connected edges was not fully considered in this process. For the above reasons, these methods could not provide accurate central nodes and use these relationships to mine for more implicit information. While dividing the network to obtain the nodes' situation more specifically in each community, different noncentral nodes need to be divided into each central node to form the communities. As a result, the central nodes of the whole network must be located quickly and precisely. In this paper, the diffusion kernel similarity [40] is used to measure the relationship between nodes and thus determine the central nodes, which is calculated as follows:

$$S = e^{-\gamma L} \tag{1}$$

where γ should be set to 1 and L is the Laplacian matrix. The bigger the diffusion kernel similarity between nodes in the diffusion kernel similarity matrix is, the closer the relationship between nodes is. Conversely, the sparser relationship is.

According to the FCMdd algorithm [41], i.e., the central nodes of the network were the center of each community. However, before dividing the network, the initial communities need to be identified based on the connected edges' relationship between the nodes to find the central nodes. In the initial stage of network detection in order to make full use of the similarity relationship between nodes, and to facilitate the setting of similarity communities, this paper proposes the similarity threshold for node i (All node i involves in this paper belong to the following range $i = 1, 2, \ldots, n$, n is the number of nodes in the network), which is calculated as follows:

$$S_{thre_i} = \frac{\sum_{j=1}^{n} S(i,j)}{n} \tag{2}$$

where $S(i,j)$ is the diffusion kernel similarity between node i and node j, and n is the number of nodes in the network.

After setting the similarity threshold S_{thre}, the similarity situation between the nodes needs to be determined based on the relationship between the diffusion kernel similarity and the similarity threshold of each node. Therefore, the determination method of the similarity community C_{simi} is proposed in this paper, and the equation is as follows:

$$C_{simi_i} = \begin{cases} C_{simi_i} \cup j, & \text{if } S(i,j) > S_{thre_i} \\ C_{simi_i}, & else S(i,j) \le S_{thre_i} \end{cases} \tag{3}$$

where C_{simi_i} is the similarity community of node i. When $S(i,j)$ is greater than the set similarity threshold S_{thre_i}, it means that from the definition of similarity, node j should belong to the similarity community of node i, i.e., C_{simi_i}. Otherwise, C_{simi_i} remains unchanged. Eq. (3) can fully use the relationship between the diffusion kernel similarity of each node and the similarity threshold, then determine the implied similarity community by simple network information. Establishing the similarity community also facilitates the precise selection of the central nodes.

As shown above, this paper finds the central nodes as follows. The nodes are selected according Eq. (3) to obtain the similarity community C_{simi}. Then the central nodes are obtained by the diffusion kernel similarity between each node in each similarity community C_{simi}. Fig. 2 shows the process of dividing the similarity communities and finding the central nodes.

In Fig. 2, first, the diffusion kernel similarity matrix S is calculated according to Eq. (1), then similarity threshold S_{thre} of each node is obtained in combination with Eq. (2). Second, in the diffusion kernel similarity matrix S, the similarity communities C_{simi} are divided for each node according to Eq. (3), and the similarity communities of each blue node are given in the figure, respectively. Last, the similarity communities are summed and sorted by the diffusion kernel similarity in descending order. Then nodes in the sort are taken out in turn as the central nodes, and the adjacent nodes of their similarity communities are deleted. The central nodes' set of the network can be obtained as red nodes.

Algorithm 1 is the procedure of the community initialization based on the diffusion kernel similarity.

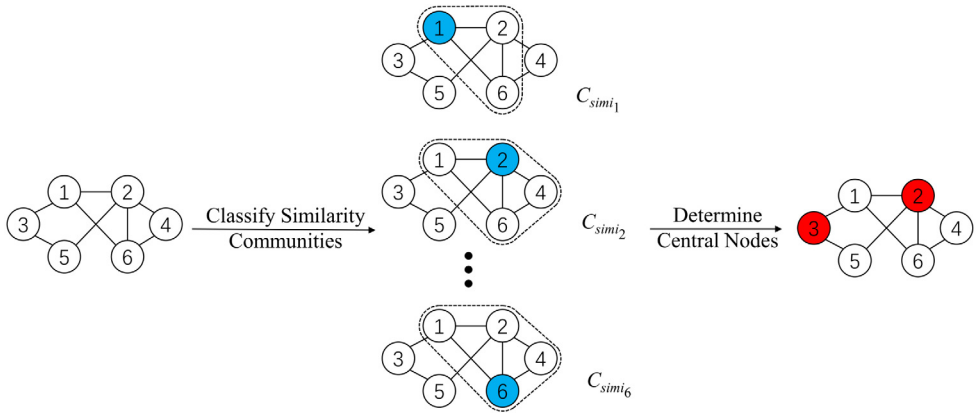

Fig. 2. The process of finding central nodes based on diffusion kernel similarity.

Algorithm 1 The community initialization based on the diffusion kernel similarity

Input: adjacency matrix A_{mat}, number of nodes n, number of populations N.
Output: diffusion kernel similarity matrix S, similarity community C_{simi}, initial population POP.

Initialization: calculate S by (1) and the similarity threshold S_{thre} by (2);
for $i = 1; i < n; i++$ **do**
  for $j = 1; j < n; j++$ **do**
    $C_{simi_i} \leftarrow$ The similarity community of node i is obtained by (3);
  end for
end for
$S_{simi} \leftarrow$Sum the diffusion kernel similarity of each C_{simi} and sort them in descending order;
$node_{all} \leftarrow$Recording the node set of S_{simi};
while $size(node_{all}) \neq null$ **do**
  $node_{cn} \leftarrow$ Select nodes from $node_{all}$ and delete the adjacent nodes according to its similarity community;
end while
for $m = 1; m < N; m++$ **do**
  Randomly select $2 \sim size(node_{cn})$ nodes from $node_{cn}$;
end for
return S, C_{simi} and POP.

2.2. Pre-division based on similarity communities

After selecting the central nodes, existing methods usually only used the central nodes for nonoverlapping community detection. Therefore, these methods led to a time-consuming network initialization in the subsequent detection process. This paper proposes a pre-division method based on the similarity community to fully consider the edge-connected information of nodes and speed up the process of nonoverlapping community detection. NSGA-II [42] includes fast non-dominated sorting, crowding distance sorting, and elite retention strategy, which can obtain the nondominated frontier and the solution sets with good convergence. Therefore, under the framework of NSGA-II, the similarity communities of each central node are used as the initial communities in the process of pre-division. Two objective functions KKM [43] and RC [44], are minimized to complete continuous optimization of detection results, where KKM is the internal connection density of each community and RC is the connection density between different communities.

In [43], it was stated that the smaller values of KKM indicated a higher density of connectivity in communities. In comparison, the smaller values of RC indicated a lower density of connectivity between communities. To achieve the purpose of community detection, i.e., the connectivities within communities are tight, but the connectivities between communities are sparse. Therefore, both KKM and RC objective functions need to be minimized. The objective functions are specified as follows:

$$\min \begin{cases} \text{KKM} = 2(n-k) - \sum_{q=1}^{t} \frac{L(C_q, C_q)}{|C_q|} \\ \text{RC} = \sum_{q=1}^{t} \frac{L(C_q, \overline{C_q})}{|C_q|} \end{cases} \tag{4}$$

where n is the number of nodes in the network, k is the number of communities in the network, and C_q represents the qth community, $L(C_q, C_q) = \sum_{i \in C_q} \sum_{j \in C_q} A_{mat_{ij}}$, $L(C_q, \overline{C_q}) = \sum_{i \in C_q} \sum_{j \in \overline{C_q}} A_{mat_{ij}}$, and A_{mat} is the adjacency matrix of the network.

In the framework of NSGA-II, the central nodes are continuously optimized. After that, the number of communities can be determined automatically. However, the obtained results of community detection are nonoverlapping. And it is necessary to divide the overlapping nodes and thus get the overlapping communities. This paper uses the second stage of [45] as the final partition for dividing overlapping communities.

Following is the procedure to obtain the pre-division by the similarity communities based on the initial population POP, as shown in Algorithm 2.

Algorithm 2 The pre-division based on the initialization of the diffusion kernel similarity

Input: similarity community C_{simi}, number of populations N, initial population POP, number of iterations Gen, mutation operation $Mutation$, crossover operation $Crossover$.
Output: populations of pre-division POP_{pre}.

for $k = 1; k < N; k++$ **do**
  The initial communities$\leftarrow$Select C_{simi} of each central node and set the unique label $label = \{1, 2, \ldots, q\}$, with q being the number of central nodes;
  Calculate the KKM and RC of POP_k and each child$\leftarrow$Obtain the children of POP_k using $Mutation$ and $Crossover$;
  Use NSGA-II to get the new POP_k;
end for
$POP_{pre} = \{POP_1, POP_2, \ldots, POP_k\}$;
return POP_{pre}.

2.3. The correction strategy of noncentral nodes based on similarity community

In the pre-division method, there is the case that the noncentral nodes are incorrectly divided. However, in the method of

initializing community-based on diffusion kernel similarity, each node can judge the similarity community it belongs to by diffusion kernel similarity. Therefore, the correction of each community can be completed by the community's situation of adjacent nodes in the similarity community. To sum up the above, this paper proposes a similarity community-based correction strategy to correct noncentral nodes, and the correction equation proposed in this paper is as follows:

$$label_i = \begin{cases} label_{simi_i}, & if\ \Delta Q_{ov} > 0 \\ label_i, & else \Delta Q_{ov} \le 0 \end{cases} \tag{5}$$

where $label_i$ denotes the label of node i and $label_{simi_i}$ denotes the labels of adjacent nodes in the similarity community of node i.

Under the judgment of Eq. (5), the correction of noncentral nodes in the network can be completed. The correction strategy of noncentral nodes for the whole network based on the similarity communities is shown in Fig. 3.

In Fig. 3, for the blue node. First, each adjacent node in similarity community C_{simi_1} of the blue node is found, then the blue node's label is changed to the label of adjacent node $label_{simi_2}$ and $label_{simi_6}$, respectively. Then, according to Eq. (5) to decide whether to correct the label. Finally, the community label of the blue node is output after the correction.

The procedure of the correction strategy is as shown in Algorithm 3.

Algorithm 3 The correction strategy of the noncentral nodes based on the similarity community

Input: similarity community C_{simi}, number of populations N, number of nodes n, pre-divided populations POP_{pre}.
Output: corrected population $POP_{correct_1}$.

for $k = 1; k < N; k++$ **do**
  for $i = 1; i < n; i++$ **do**
    Select adjacent nodes of node i from C_{simi_i};
    $label_{simi} = \{label_{simi_1}, label_{simi_2}, \ldots, label_{simi_q}\}$ ← Get the community labels of the adjacent nodes, with q being the number of adjacent nodes;
    for $j = 1; j < q; j++$ **do**
      Set the label of node i to $label_{simi_n}$;
      Determine the correction according to equation (5);
    end for
  end for
end for
return $POP_{correct_1}$.

2.4. The correction strategy of overlapping nodes based on similarity community and separation operation

After correcting noncentral nodes in pre-division, the final partition is used to get the overlapping communities. Because the core of the final partition lies in discovering the overlapping nodes. Therefore, the overlapping nodes in the final division need to be corrected.

In order to utilize the distribution of the similarity communities to which overlapping nodes belong, the following adjustment is required. According to the number of communities to which overlapping nodes belong, the algorithm can select the same number of communities of adjacent nodes among the similarity communities of overlapping nodes. Subsequently, the occurrences of the adjacent nodes' communities are sorted in descending order and selected. Based on the above analysis, the overlapping nodes' correction equation proposed in this paper is as follows:

$$label_{simi-ov_j} = \arg\max_r(sort(label_{simi_i})) \tag{6}$$

where $label_{simi_j}$ is the adjacent nodes' label in the similarity community of the overlapping node j, $label_{simi-ov_j}$ is the set of label corrections for overlapping node j. Thus, Eq. (6) represents a statistical count of $label_{simi_j}$ to obtain the class of labels that makes the largest count, denoted as r.

Based on the above correction strategy, in order to make further correction to the overlapping nodes. Therefore, the separation operation can then be decided by separating overlapping nodes and judging the situation of ΔQ_{ov}. This paper selects the overlapping communities composed of blue and pink communities for the separation operation, as shown in Fig. 4.

Algorithm 4 Overlapping nodes' correction strategy based on the similarity community and the separation operation

Input: similarity community C_{simi}, number of populations N, number of overlapping nodes n_{ov}, final population of the final partition POP_{last}.
Output: corrected populations $POP_{correct_2}$.

for $k = 1; k < N; k++$ **do**
  for $j = 1; j < n_{ov}; j++$ **do**
    Select adjacent nodes of overlapping node j from C_{simi_j};
    $label_{simi}$ ← Put the community labels of the adjacent nodes;
    The number of each type of community in the $label_{simi}$ is counted and sorted in descending order;
    $label_{simi-ov}$ ←According to the number of labels of overlapping node j and equation (6) to obtain the set of label corrections;
    Set the community label of overlapping node j to the label appearing in $label_{simi-ov}$;
    if $\Delta Q_{ov} > 0$ **then**
      The overlapping node j is judged to successfully corrected;
    else
      The label of overlapping node j is set to the original label;
    end if
    Set the label of overlapping node j to its individual labels in turn;
    if $\Delta Q_{ov} > 0$ **then**
      Judge the successful separation of overlapping node j and select the label with the largest increase in extended modularity Q_{ov};
    else
      Set the label of overlapping node j to its original label.
    end if
  end for
end for
return $POP_{correct_2}$.

In Fig. 4, the purple overlapping node is considered as a nonoverlapping in the blue and pink community. After that, Q_{ov} is calculated before and after each case. If $\Delta Q_{ov} > 0$, the overlapping node is separated. Otherwise, the overlapping node will not be separated. Finally, based on the judgment of ΔQ_{ov}, the overlapping node is obtained to belong to the blue community.

The final partition's correction strategy is as shown in Algorithm 4.

2.5. Overall procedure of the algorithm

With the integration of the previous four parts, the overall procedure of the evolutionary multiobjective overlapping community detection algorithm based on similarity matrix and node correction is represented in Algorithm 5.

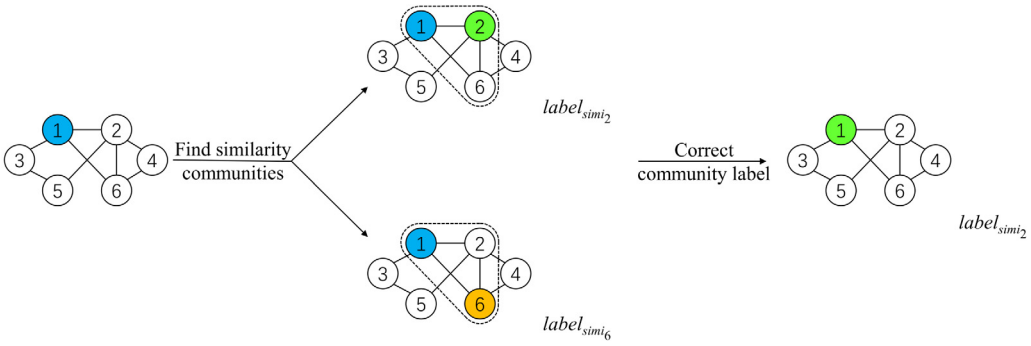

Fig. 3. The process of correcting noncentral node.

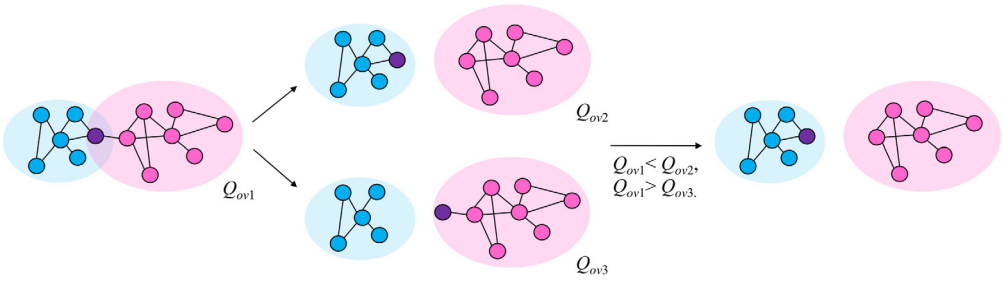

Fig. 4. The correction strategy's separation of overlapping nodes in the final partition.

Algorithm 5 Overall procedure of the algorithm

Input: adjacency matrix A_{mat}, number of nodes n, number of populations N, number of iterations Gen, mutation operation $Mutation$, crossover operation $Crossover$.

Output: the corrected nonoverlapping and overlapping results.

```
while Gen ≠ 0 do
    Use Algorithm 1 to initialize the communities;
    Divide the nonoverlapping communities by Algorithm 2;
    Gen = Gen − 1;
end while
The corrected nonoverlapping communities are obtained by Algorithm 3 to each node of the nonoverlapping communities;
Use the final partition based on AR-MOEA to detect overlapping communities;
The overlapping nodes are corrected and separated through Algorithm 4 to obtain the corrected overlapping communities;
return the corrected nonoverlapping and overlapping results.
```

2.6. Time complexity analysis

Suppose that the number of nodes and edges in graph G are n and m, respectively. N is the population size in nonoverlapping partition, N' is the population size in overlapping partition, and Gen is the number of iterations. Then the time complexity of the proposed algorithm consists of five components: Algorithm 1 is $O(n^2+n+N)$, Algorithm 2 is $O((n+m+N^2)\times N)$, Algorithm 3 is $O(N\times n^2)$, Algorithm 4 is $O(N\times n)$, and the time complexity at the second stage of [45]. The total time complexity of the proposed algorithm is finally obtained according to Algorithm 5. Since the number of nodes is much larger than the population size of two partitions, the total time complexity of the proposed algorithm can be simplified to $O(n^2\times N\times N'\times Gen)$.

3. Experiment and analysis

The experimental part is implemented on matlab2020a software. The processor is Intel(R) Core(TM) i5-8250U CPU @ 1.60 GHz, the memory is 16.0 GB, and the operating system is Windows 10. In the pre-division method, variation operation and crossover operation are performed by the bitwise variation and the uniform crossover, respectively. Population size and the maximum number of generations are set to 100.

3.1. Evaluation metrics

The generalized normalized mutual information (gNMI) [23] is the first evaluation metric to verify the algorithm's performance. This evaluation metric compares the real partitions of the network with the algorithm's detection results and can only be used in networks where the real partition exists. gNMI is defined as:

$$\mathrm{gNMI}(A,B)=\frac{-2\sum_{i=1}^{K_A}\sum_{j=1}^{K_B}C_{ij}\log(C_{ij}\cdot n/C_{i.}C_{.j})}{\sum_{i=1}^{K_A}C_{i.}\log(C_{i.}/n)+\sum_{j=1}^{K_B}C_{.j}\log(C_{.j}/n)} \tag{7}$$

where n denotes the total number of nodes in the network, K_A and K_B denote the number of real communities and the number of communities detected by algorithm, respectively, C_{ij} denotes the number of the same nodes between the community i in A and the community j in B, and $C_{i.}(C_{.j})$ denotes the sum of elements in row i (column j) of C. The value of gNMI(A,B) ranges from [0, 1], and the higher value of gNMI(A,B) indicates that the effect of community detection is better. Some clarifications are provided below. If gNMI$(A,B)=0$, it indicates that A and B are completely different, whereas gNMI$(A,B)=1$, it signifies that A and B are identical, i.e., the algorithm detects the true community structure of the network.

The second evaluation metric is Q_{ov} [46], which measures the difference between the number of edges in the given community and the expected number when edges are randomly distributed.

Table 1
Specific parameters of the LFR networks.

Network name	N	μ	O_n	O_m	C_{min}	C_{max}	Others
LFR1	100	{0.1,0.2,0.3,0.4,0.5}	0.1N	{2,4,6,8}	10	20	$k = 10$
LFR2	500	{0.1,0.2,0.3,0.4,0.5}	0.1N	{2,4,6,8}	30	60	$k_{max} = 40$
LFR3	1000	{0.1,0.2,0.3,0.4,0.5}	0.1N	{2,4,6,8}	30	60	$\tau_1 = 2$, $\tau_2 = 1$

Obviously, Q_{ov} can be used when the real community structure is unknown. Because of this, the variety of networks used for detection is greatly increased. Q_{ov} is defined as:

$$Q_{ov} = \frac{1}{2m}\sum_{q=1}^{l}\sum_{i\in C_q, j\in C_q}\frac{1}{O_i O_j}\left(A_{mat_{ij}} - \frac{k_i k_j}{2m}\right) \tag{8}$$

where m is the number of edges in the network, O_i is the number of communities to which node i belongs, and k_i is the degree of node i.

3.2. Comparison algorithms

This paper adopts the following four algorithms for comparison, the algorithm based on fuzzy methods (EMOFM-DK) [45], the local spectral diffusion based approach (LEMON) [47], the algorithm based on seed expansion (NISE) [48], and the Bayesian nonnegative matrix factorization based approach (NMF) [30]. In this paper, the experimental parameters in both synthetic networks as well as real networks are set according to the parameters recommended by the four algorithms.

3.3. Experimental results on the LFR networks

3.3.1. LFR networks

Lancichinetti et al. proposed the LFR network [49], and this paper uses the LFR model to generate different synthetic networks. Some adjustable parameters are included in the LFR network to control the network's structure: N is the number of nodes in the network, i.e., the size of the network; k and k_{max} are the average node degree and the maximum node degree of the network, respectively; τ_1 and τ_2 are the exponents of power-law distribution followed by node degree and community size, respectively; μ is to control the degree of connectivity between communities; O_n is the number of overlapping nodes in the network; O_m is the number of communities to which each overlapping node belongs; C_{min} and C_{max} are the minimum and maximum size of each community, respectively.

This paper sets the size N of the LFR networks to 100, 500, and 1000. Subsequently, these LFR networks are denoted as LFR1, LFR2, and LFR3. The community size of LFR1 is $[C_{min}, C_{max}] = [10,20]$, and the community size of LFR2 and LFR3 is $[C_{min}, C_{max}] = [30,60]$. The other parameters of the three LFR networks are set identically: k and k_{max} are set to 10 and 40, respectively. τ_1 and τ_2 are set to 2 and 1, respectively. μ is varied from 0.1 to 0.5 in steps of 0.1. O_n is set to 0.1N. O_m is varied from 2 to 8 in steps of 2. The specific parameters of the three LFR networks are listed as shown in Table 1.

3.3.2. The results and analysis of LFR networks

For the three different LFR networks, LFR1, LFR2, and LFR3, each algorithm is run 20 times to obtain community results. Since LFR networks' corresponding real structures are obtained when they are generated, the experimental results for LFR networks are measured using gNMI and Q_{ov} metrics.

Fig. 5 shows the gNMI for the results of community detection as μ changes from 0.1 to 0.5 for LFR1 with O_m set to 2, 4, 6, and 8, respectively.

In Fig. 5(a), when $0.1 \le \mu \le 0.3$, the gNMI values of the proposed algorithm and EMOFM-DK always remain above 0.85, and the gNMI values of NMF can stay above 0.75. The gNMI values of these three algorithms can maintain certain stability in this interval. However, after $\mu > 0.3$, the gNMI values of NMF decrease rapidly and drop to about 0.1 at $\mu = 0.5$. At the same time, the gNMI values of the proposed algorithm and EMOFM-DK decrease but remain about 0.5 at $\mu = 0.5$. In Fig. 5(b), the gNMI values of the proposed algorithm and EMOFM-DK always remain above 0.65 when $0.1 \le \mu \le 0.3$, and the gNMI values of EMOFM-DK rise back to about 0.7 when $\mu = 0.3$. However, after $\mu > 0.3$, the gNMI values of the proposed algorithm and EMOFM-DK decrease, and only the proposed algorithm remains around 0.5 at $\mu = 0.5$. In Fig. 5(c) and (d), the gNMI values of the proposed algorithm always stay above 0.5 throughout the change of μ, which can ensure good stability. In summary, in the LFR1 network, the gNMI values of the proposed algorithm have a good advantage. And the stability of the proposed algorithm can be better and better as the overlap of the network gradually deepens. For the other three algorithms, they cannot guarantee stability.

Fig. 6 shows the gNMI for the LFR2 network. And the change process of μ and O_m is consistent with the LFR1.

Corresponding to Fig. 6, the LFR2 network's size is 500, and the proposed algorithm show good stability in Fig. 6(c) and (d). In Fig. 6(a), all algorithms show a similar decreasing trend after $\mu > 0.3$, while the proposed algorithm and EMOFM-DK can maintain some stability before that. In Fig. 6(b), when $0.1 \le \mu \le 0.3$, the gNMI values of the proposed algorithm, EMOFM-DK, and NMF all remain above 0.5. And as the relationship between communities is gradually blurred, the gNMI values of most algorithms first decrease steadily, and the gNMI values of all algorithms decrease rapidly after μ exceeded 0.4. In Fig. 6(c), as μ increases to 0.3, the gNMI values of the proposed algorithm, EMOFM-DK, and NMF can remain around 0.5. However, as the relationships within the community become more sparse, only the gNMI values of the proposed algorithm and EMOFM-DK show some decrease and remain above 0.5. In Fig. 6(d), the proposed algorithm is more stable throughout the change of μ, and the gNMI values always stay around 0.5. It can be seen that the proposed algorithm has better overall results in the LFR2 network and shows better stability at $O_m = 8$. EMOFM-DK also achieves good results in most cases. However, NMF only achieves good values of gNMI and stability when μ is small.

Fig. 7 shows the gNMI for the LFR3. And the change process of μ and O_m is consistent with the LFR1 and LFR2.

Corresponding to Fig. 7, since the size of the LFR3 network is 1000 at this time, it is very comparatively difficult to maintain certain stability within the same distribution of O_n and the variation range of μ as LFR1 and LFR2. The gNMI values of all algorithms basically decrease with increasing μ, and none can maintain stability. And only the NMF's values of gNMI rebound in a few cases with $\mu = 0.3$ in Fig. 7(b) and (d). From the comprehensive analysis of the LFR3 network, the overall results of the proposed algorithm are better, while EMOFM-DK and NMF can also achieve good results in most cases.

Q_{ov} is then used to measure the results of community detection as μ changes from 0.1 to 0.5 for each LFR network with O_m set to 2, 4, 6, and 8, respectively, as shown in Fig. 8.

As can be seen in the various subplots in Fig. 8. In the LFR1 network, the Q_{ov} of the proposed algorithm and EMOFM-DK can achieve better results, and the results of the proposed algorithm are more stable when O_m is 4, 6, and 8, respectively.

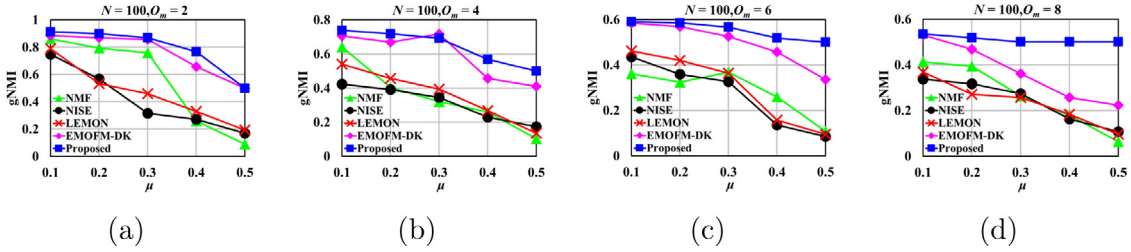

Fig. 5. gNMI values for the LFR1 network.

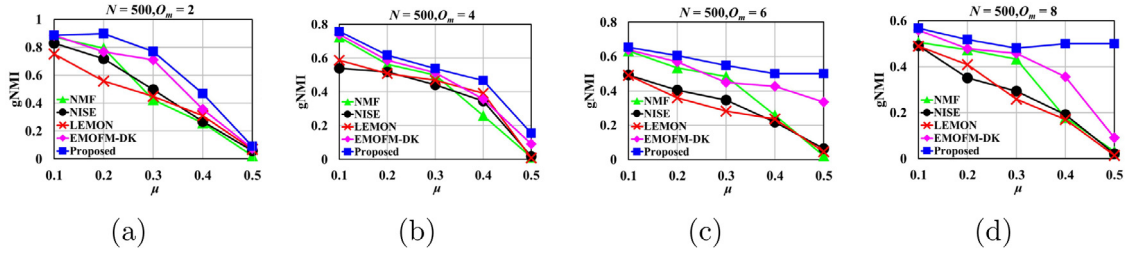

Fig. 6. gNMI values for the LFR2 network.

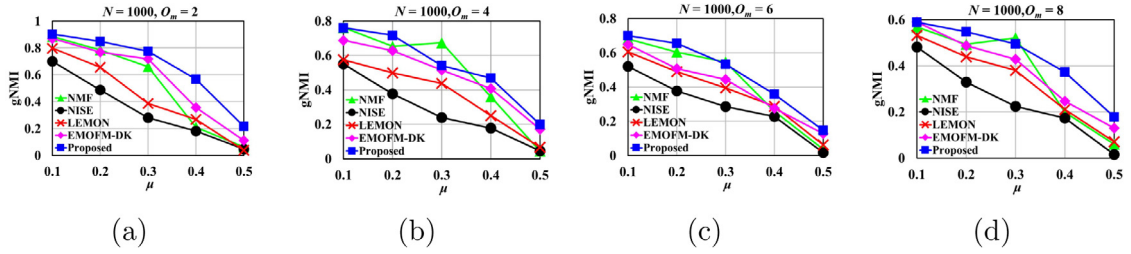

Fig. 7. gNMI values for the LFR3 network.

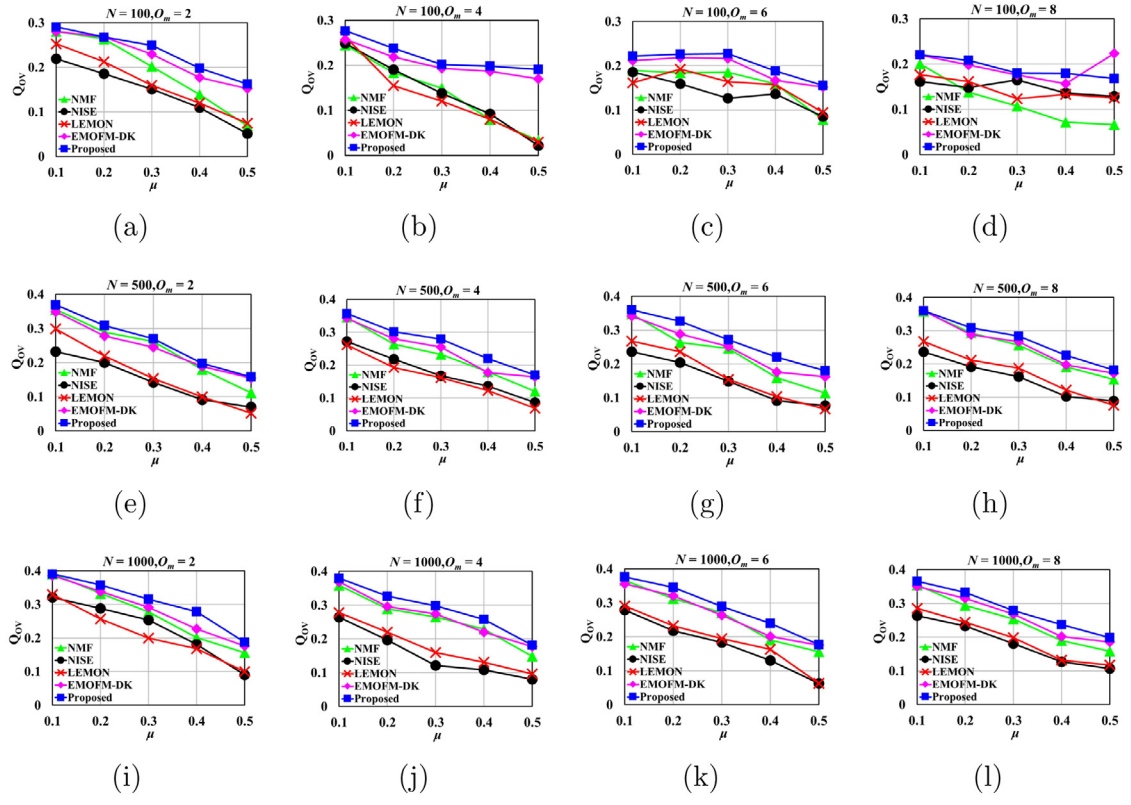

Fig. 8. Q_{ov} values for LFR1, LFR2, and LFR3 networks.

Table 2
Specific information of the real networks.

Network	Node	Edge	Average degree	Real clusters	Reference
ENZYMES_g163	12	22	3.67	Unknown	[50]
Karate	34	78	4.59	2	[51]
Dolphin	62	159	5.13	2	[52]
Polbook	105	441	8.4	3	[53]
Football	115	613	10.66	12	[53]
SFI	118	200	3.39	Unknown	[44]
Jazz	198	2742	27.70	Unknown	[54]
Gene-fusion	291	279	1.92	Unknown	[55]
Celegansmetabolic	453	2025	8.94	Unknown	[56]
Email	1133	5451	9.62	Unknown	[38]
Yeast-D2	1443	6993	9.69	162	[57]
Netscience	1589	2742	3.45	Unknown	[44]
Y2H	1966	2705	2.75	203	[58]
ego-Facebook	4039	88 234	43.69	Unknown	[59]
Powergrid	4941	6594	2.67	Unknown	[7]
Erdos	6927	11 850	3.42	Unknown	[56]
Lastfm-asia	7624	27 806	7.29	Unknown	[60]

Comprehensive analysis of LFR2 and LFR3 networks, as the fuzzy degree of the network structure deepens, the Q_{ov} of the proposed algorithm, EMOFM-DK, and NMF can achieve better results.

Through the above analysis of gNMI and Q_{ov} metrics for each LFR network, the following conclusions can be synthesized. In terms of the accuracy of the metric results, although the proposed algorithm, EMOFM-DK, and NMF can achieve higher accuracy in most cases, the proposed algorithm can basically achieve the highest accuracy metric results. In terms of stability, the proposed algorithm can maintain good stability of the results in many cases. Therefore, the proposed algorithm can maintain a better advantage in terms of the accuracy of results and the stability.

3.4. Experimental results on the real networks

3.4.1. Real networks

Seventeen real networks are used in this experiment, six of which have real clusters. The specific information of real networks is shown in Table 2.

3.4.2. The results and analysis of real networks

In this section, the networks' detection results of the proposed algorithm and all comparison algorithms are measured using gNMI and Q_{ov}. The maximum value, average value and standard deviation of each metric are listed. The experimental results are obtained by running each algorithm independently 20 times, and "—" indicates that the algorithm cannot give the networks' detection results in effective time. Table 3 lists the values of gNMI for six networks with the real community structures. In comparison, Table 4 lists the values of Q_{ov} for all seventeen networks.

From Table 3, it can be seen that the proposed algorithm can achieve the best detection results in both the maximum and average values of gNMI in six real networks. The proposed algorithm and EMOFM-DK can correctly divide the Karate network, i.e., gNMI_max = gNMI_avg = 1. For the Dolphin network, both the proposed algorithm and EMOFM-DK's the maximum value of gNMI is 1. Still, the overall operation of the proposed algorithm is more stable. The advantage of the proposed algorithm is obvious from the maximum and average results of gNMI for Polbook and Football networks. For the Yeast-D2 network, although both the proposed algorithm and EMOFM-DK achieve 0.5401 for the maximum value of gNMI, the proposed algorithm increases the average gNMI value from 0.3137 to 0.3565. Therefore, the proposed algorithm is significantly better than other comparative algorithms in terms of networks' division for the six known communities.

Table 3
The values of gNMI detected by five algorithms on six real networks.

	Metric	NMF	NISE	LSC	EMOFM-DK	Proposed
Karate	gNMI_max	0.4067	0.8887	0.9214	**1.0000**	**1.0000**
	gNMI_avg	0.3130	0.8887	0.8998	**1.0000**	**1.0000**
	gNMI_std	0.0831	0.0000	0.0382	0.0000	0.0000
Dolphin	gNMI_max	0.3121	0.6647	0.7343	**1.0000**	**1.0000**
	gNMI_avg	0.2662	0.6647	0.7328	0.9889	**0.9963**
	gNMI_std	0.0458	0.0000	0.0017	0.0339	0.0203
Polbook	gNMI_max	0.1562	0.4015	0.3552	0.3587	**0.5000**
	gNMI_avg	0.1375	0.4015	0.3530	0.3497	**0.4895**
	gNMI_std	0.0183	0.0000	0.0004	0.0076	0.0072
Football	gNMI_max	0.7947	0.7729	0.7583	0.8203	**0.8458**
	gNMI_avg	0.6899	0.7729	0.7534	0.8126	**0.8329**
	gNMI_std	0.1064	0.0000	0.0047	0.0052	0.0114
Yeast-D2	gNMI_max	0.1580	0.1209	0.2067	**0.5401**	**0.5401**
	gNMI_avg	0.1398	0.1209	0.2024	0.3137	**0.3565**
	gNMI_std	0.0164	0.0000	0.0036	0.1375	0.1283
Y2H	gNMI_max	0.0249	0.0734	0.0792	0.0881	**0.0930**
	gNMI_avg	0.0234	0.0734	0.0767	0.0851	**0.0922**
	gNMI_std	0.0013	0.0000	0.0025	0.0026	0.0006

Table 4 shows that the proposed algorithm can achieve the best results of Q_{ov} among the seventeen networks. It can be seen that when the size of the network is small, the difference in the detection results of Q_{ov} between the proposed algorithm and the comparison algorithms is slight. But as the networks' size increases, the proposed algorithm obtains the greater results of Q_{ov} in nine networks: Email, ego-Facebook, SFI, Netscience, Celegansmetabolic, Yeast-D2, Gene-fusion, Powergrid, and Lastfm-asia, and at least 1% better than the best results in the comparison algorithms. Therefore, from the Q_{ov} metric, the proposed algorithm is more competitive.

The above experiments on real networks show that the proposed algorithm can achieve the best detection results under the two metrics of gNMI and Q_{ov}. On some small networks, the optimal value of Q_{ov} is limited by the size of the networks, which leads to no effective improvement in the best results of the proposed algorithm compared to the comparison algorithms. But with the increasing of network size, the proposed algorithm can effectively enhance the value of Q_{ov} in nine networks. Therefore, the proposed algorithm can achieve more apparent advantages and good stability.

4. Conclusion and future work

This paper proposes an evolutionary multiobjective overlapping community detection algorithm based on similarity matrix and node correction. The proposed algorithm can solve the problem of the high sensitivity of the community centers' initialization in overlapping community detection algorithms based on fuzzy clustering and correct the misclassified nodes. To solve the above problems, the proposed algorithm proposes the concept of similarity community based on the diffusion kernel similarity to fully use the relationship of the connected edges between nodes to find the central nodes. And different correction strategies based on the similarity communities are designed for the noncentral nodes and the overlapping nodes misclassified in the detection process. In this paper, four excellent overlapping community detection algorithms in recent years are used as comparisons on three different sizes of synthetic networks. In terms of the gNMI metric, the proposed algorithm can achieve significant advantages in most cases and certain stability in the LFR1 network. In terms of the Q_{ov} metric, the proposed algorithm achieves a good advantage in terms of metric values and stability. In addition, all algorithms partition the seventeen real networks and use gNMI and Q_{ov}

Table 4
The values of Q_{ov} detected by the five algorithms on seventeen real networks.

	Metric	NMF	NISE	LSC	EMOFM-DK	Proposed
ENZYMES_g163	Q_{ov}_max	0.2567	0.2479	0.2411	**0.2567**	**0.2567**
	Q_{ov}_avg	0.2488	0.2479	0.2411	**0.2567**	**0.2567**
	Q_{ov}_std	0.0015	0.0000	0.0000	0.0000	0.0000
Karate	Q_{ov}_max	0.1314	0.1532	0.2317	**0.2348**	**0.2348**
	Q_{ov}_avg	0.0727	0.1532	0.2317	0.2341	**0.2348**
	Q_{ov}_std	0.0538	0.0000	0.0000	0.0003	0.0000
Dolphin	Q_{ov}_max	0.2565	0.2271	0.2538	0.2730	**0.2750**
	Q_{ov}_avg	0.2404	0.2271	0.2487	0.2723	**0.2743**
	Q_{ov}_std	0.0137	0.0000	0.0045	0.0004	0.0003
Polbook	Q_{ov}_max	0.2673	0.2163	0.2592	**0.2704**	**0.2704**
	Q_{ov}_avg	0.2655	0.2163	0.2586	**0.2702**	**0.2702**
	Q_{ov}_std	0.0014	0.0000	0.0004	0.0002	0.0001
Football	Q_{ov}_max	0.3059	0.2466	0.2851	0.3066	**0.3067**
	Q_{ov}_avg	0.3039	0.2466	0.2735	0.3063	**0.3066**
	Q_{ov}_std	0.0016	0.0000	0.0114	0.0003	0.0000
Email	Q_{ov}_max	0.2744	0.2267	0.2454	0.2735	**0.2851**
	Q_{ov}_avg	0.2702	0.2267	0.2341	0.2728	**0.2832**
	Q_{ov}_std	0.0035	0.0000	0.0107	0.0005	0.0009
ego-Facebook	Q_{ov}_max	0.4153	0.3995	0.4075	0.4012	**0.4127**
	Q_{ov}_avg	0.4149	0.3995	0.4052	0.3988	**0.4106**
	Q_{ov}_std	0.0003	0.0000	0.0012	0.0027	0.0018
SFI	Q_{ov}_max	0.3764	0.3414	0.3646	0.3731	**0.3837**
	Q_{ov}_avg	0.3764	0.3414	0.3617	0.3716	**0.3821**
	Q_{ov}_std	0.0000	0.0000	0.0028	0.0013	0.0015
Jazz	Q_{ov}_max	0.2194	0.1398	0.1863	0.2259	**0.2261**
	Q_{ov}_avg	0.2179	0.1398	0.1825	0.2258	**0.2260**
	Q_{ov}_std	0.0012	0.0000	0.0036	0.0001	0.0001
Netscience	Q_{ov}_max	0.4565	0.3649	0.3791	0.4632	**0.4748**
	Q_{ov}_avg	0.4562	0.3649	0.3742	0.4624	**0.4731**
	Q_{ov}_std	0.0002	0.0000	0.0043	0.0006	0.0013
Erdos	Q_{ov}_max	0.2223	0.2372	0.2804	0.3255	**0.3299**
	Q_{ov}_avg	0.2062	0.2372	0.2779	0.3227	**0.3268**
	Q_{ov}_std	0.0158	0.0000	0.0021	0.0024	0.0026
Celegansmetabolic	Q_{ov}_max	0.1928	0.1833	0.1978	0.2105	**0.2242**
	Q_{ov}_avg	0.1871	0.1833	0.1947	0.2101	**0.2231**
	Q_{ov}_std	0.0043	0.0000	0.0026	0.0002	0.0009
Yeast-D2	Q_{ov}_max	0.4175	0.3032	0.3389	0.4142	**0.4263**
	Q_{ov}_avg	0.4172	0.3032	0.3346	0.4125	**0.4232**
	Q_{ov}_std	0.0002	0.0000	0.0028	0.0013	0.0025
Y2H	Q_{ov}_max	0.0539	0.1125	0.1183	0.3576	**0.3691**
	Q_{ov}_avg	0.0511	0.1125	0.1154	0.3559	**0.3674**
	Q_{ov}_std	0.0024	0.0000	0.0026	0.0015	0.0013
Gene-fusion	Q_{ov}_max	–	–	0.3732	0.4123	**0.4302**
	Q_{ov}_avg	–	–	0.3694	0.4089	**0.4295**
	Q_{ov}_std	–	–	0.0027	0.0036	0.0004
Powergrid	Q_{ov}_max	0.3857	0.2265	0.3067	0.4458	**0.4581**
	Q_{ov}_avg	0.3845	0.2265	0.2958	0.4377	**0.4526**
	Q_{ov}_std	0.0006	0.0000	0.0009	0.0055	0.0052
Lastfm-asia	Q_{ov}_max	–	0.2073	0.2275	0.3853	**0.3968**
	Q_{ov}_avg	–	0.2073	0.2221	0.3836	**0.3942**
	Q_{ov}_std	–	0.0000	0.0032	0.0012	0.0034

metrics. EMOFM-DK can achieve good detection results of gNMI and Q_{ov} on many networks. Meanwhile, the proposed algorithm can achieve the best gNMI on six networks with real clusters. And in Email, ego-Facebook, SFI, Netscience, Celegansmetabolic, Yeast-D2, Gene-fusion, Powergrid, and Lastfm-asia networks, the proposed algorithm's division results of Q_{ov} achieve a significant advantage. The concept of similarity communities proposed in this paper, it can find more accurate central nodes and thus speed up the process of obtaining nonoverlapping communities under the evolutionary multiobjective algorithm. However, the problem of the long-running time of the evolutionary algorithm still needs to be solved. In future work, parallel procedures can be used to apply the proposed algorithm to more large-scale networks. Future work also needs further improve the proposed algorithm and find faster methods to obtain the nonoverlapping and overlapping communities.

CRediT authorship contribution statement

Ronghua Shang: Revising the article, Supervision, Project administration. **Kejia Zhao:** Design, Methodology, Investigation, Performing experiments or data/evidence collection, Drafting and revising the article. **Weitong Zhang:** Revising the article, Supervision, Project administration. **Jie Feng:** Supervision, Project administration, Funding acquisition. **Yangyang Li:** Supervision, Project administration, Funding acquisition. **Licheng Jiao:** Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grants Nos. 62176200, 61773304, and 61871306, the Natural Science Basic Research Program of Shaanxi, China under Grant No. 2022JC-45, 2022JQ-616 and the Open Research Projects of Zhejiang Lab, China under Grant 2021KG0AB03, the 111 Project, the National Key R&D Program of China, the Guangdong Provincial Key Laboratory, China under Grant No. 2020B121201001 and the GuangDong Basic and Applied Basic Research Foundation, China under Grant No. 2021A1515110686.

References

[1] R. Shang, W. Zhang, L. Jiao, X. Zhang, R. Stolkin, Dynamic immunization node model for complex networks based on community structure and threshold, IEEE Trans. Cybern. (2020).
[2] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications, Cambridge Univ. Press, Cambridge, U.K., 1994.
[3] M.E. Newman, The structure of scientific collaboration networks, Proc. Nat. Acad. Sci. USA 98 (2) (2001) 404–409.
[4] C. Pizzuti, S.E. Rombo, Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods, Bioinformatics 30 (10) (2014) 1343–1352.
[5] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Nat. Acad. Sci. USA 99 (12) (2002) 7821–7826.
[6] R. Shang, H. Liu, L. Jiao, A.M.G. Esfahani, Community mining using three closely joint techniques based on community mutual membership and refinement strategy, Appl. Soft Comput. 61 (2017) 1060–1073.
[7] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (6684) (1998) 440–442.
[8] A.L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[9] W. Zhang, R. Shang, L. Jiao, Complex network graph embedding method based on shortest path and moea/d for community detection, Appl. Soft Comput. 97 (2020) 106764.
[10] M. Gong, Q. Cai, L. Ma, L. Jiao, Big network analytics based on nonconvex optimization, in: Big Data Optimization: Recent Developments and Challenges, Springer, 2016, pp. 345–373.
[11] J.P. Bagrow, E.M. Bollt, Local method for detecting communities, Phys. Rev. E 72 (4) (2005) 046108.
[12] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Nat. Acad. Sci. USA 105 (4) (2008) 1118–1123.
[13] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008) P10008.
[14] R. Shang, W. Zhang, J. Zhang, L. Jiao, Y. Li, R. Stolkin, Local community detection algorithm based on alternating strategy of strong fusion and weak fusion, IEEE Trans. Cybern. (2022).
[15] M.E. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.
[16] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2) (2005) 027104.
[17] R. Shang, J. Bai, L. Jiao, C. Jin, Community detection based on modularity and an improved genetic algorithm, Physica A 392 (5) (2013) 1215–1231.
[18] A. Ramesh, G. Srivatsun, Evolutionary algorithm for overlapping community detection using a merged maximal cliques representation scheme, Appl. Soft Comput. 112 (2021) 107746.
[19] K. Nath, S. Roy, S. Nandi, Inovin: A fuzzy-rough approach for detecting overlapping communities with intrinsic structures in evolving networks, Appl. Soft Comput. 89 (2020) 106096.
[20] S. Gregory, Fuzzy overlapping communities in networks, J. Stat. Mech. Theory Exp. 2011 (02) (2011) P02017.
[21] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814–818.
[22] S. Gregory, A fast algorithm to find overlapping communities in networks, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 408–423.
[23] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. 11 (3) (2009) 033015.
[24] P. Zhang, J. Li, T. Li, H. Chen, A new many-objective evolutionary algorithm based on determinantal point processes, IEEE Trans. Evol. Comput. 25 (2) (2020) 334–345.
[25] Z. Gong, H. Chen, B. Yuan, X. Yao, Multiobjective learning in the model space for time series classification, IEEE Trans. Cybern. 49 (3) (2018) 918–932.
[26] H. Chen, X. Yao, Multiobjective neural network ensembles based on regularized negative correlation learning, IEEE Trans. Knowl. Data Eng. 22 (12) (2010) 1738–1751.
[27] X.N. Shen, L.L. Minku, N. Marturi, Y.N. Guo, Y. Han, A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling, Inform. Sci. 428 (2018) 1–29.
[28] X.N. Shen, X. Yao, Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems, Inform. Sci. 298 (2015) 198–224.
[29] L. Hu, K.C. Chan, Fuzzy clustering in a complex network based on content relevance and link structures, IEEE Trans. Fuzzy Syst. 24 (2) (2015) 456–470.
[30] I. Psorakis, S. Roberts, M. Ebden, B. Sheldon, Overlapping community detection using bayesian non-negative matrix factorization, Phys. Rev. E 83 (6) (2011) 066114.
[31] J. Wu, L. Jiao, C. Jin, F. Liu, M. Gong, R. Shang, W. Chen, Overlapping community detection via network dynamics, Phys. Rev. E 85 (1) (2012) 016115.
[32] W. Wang, D. Liu, X. Liu, L. Pan, Fuzzy overlapping community detection based on local random walk and multidimensional scaling, Physica A 392 (24) (2013) 6578–6586.
[33] L. Wang, Y. Liu, X. Zhao, Y. Xu, Particle swarm optimization for fuzzy c-means clustering, in: Proc. IEEE World Congr. Intell. Control Autom., Vol. 2, IEEE, 2006, pp. 6055–6058.
[34] H. Izakian, A. Abraham, Fuzzy C-means and fuzzy swarm for fuzzy clustering problem, Expert Syst. Appl. 38 (3) (2011) 1835–1838.
[35] S. Wikaisuksakul, A multi-objective genetic algorithm with fuzzy c-means for automatic data clustering, Appl. Soft Comput. 24 (2014) 679–691.
[36] T.C. Havens, J.C. Bezdek, C. Leckie, K. Ramamohanarao, M. Palaniswami, A soft modularity function for detecting fuzzy communities in social networks, IEEE Trans. Fuzzy Syst. 21 (6) (2013) 1170–1175.
[37] X. Wang, G. Liu, L. Pan, J. Li, Uncovering fuzzy communities in networks with structural similarity, Neurocomputing 210 (2016) 26–33.
[38] Z. Ding, X. Zhang, D. Sun, B. Luo, Overlapping community detection based on network decomposition, Sci. Rep. 6 (1) (2016) 1–11.
[39] A. Biswas, B. Biswas, Fuzag: Fuzzy agglomerative community detection by exploring the notion of self-membership, IEEE Trans. Fuzzy Syst. 26 (5) (2018) 2568–2577.
[40] R.I. Kondor, J. Lafferty, Diffusion kernels on graphs and other discrete structures, in: Proc. Int. Conf. Mach. Learn., Vol. 2002, 2002, pp. 315–322.
[41] R. Krishnapuram, A. Joshi, O. Nasraoui, L. Yi, Low-complexity fuzzy relational clustering algorithms for web mining, IEEE Trans. Fuzzy Syst. 9 (4) (2001) 595–607.
[42] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[43] L. Angelini, S. Boccaletti, D. Marinazzo, M. Pellicoro, S. Stramaglia, Identification of network modules by optimization of ratio association, Chaos: Interdiscip. J. Nonlinear Sci. 17 (2) (2007) 023114.
[44] M. Gong, Q. Cai, X. Chen, L. Ma, Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition, IEEE Trans. Evol. Comput. 18 (1) (2013) 82–97.
[45] Y. Tian, S. Yang, X. Zhang, An evolutionary multiobjective optimization based fuzzy method for overlapping community detection, IEEE Trans. Fuzzy Syst. 28 (11) (2020) 2841–2855.
[46] V. Nicosia, G. Mangioni, V. Carchiolo, M. Malgeri, Extending the definition of modularity to directed graphs with overlapping communities, J. Stat. Mech. Theory Exp. 2009 (03) (2009) P03024.
[47] Y. Li, K. He, K. Kloster, D. Bindel, J. Hopcroft, Local spectral clustering for overlapping community detection, ACM Trans. Knowl. Discov. Data 12 (2) (2018) 1–27.
[48] J.J. Whang, D.F. Gleich, I.S. Dhillon, Overlapping community detection using neighborhood-inflated seed expansion, IEEE Trans. Knowl. Data Eng. 28 (5) (2016) 1272–1284.

[49] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4) (2008) 046110.
[50] R. Rossi, N. Ahmed, The network data repository with interactive graph analytics and visualization, in: Proc. 29th AAAI Conf. Artif. Intell., 2015.
[51] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.
[52] D. Lusseau, The emergent properties of a dolphin social network, Proc. R. Soc. Lond. B Biol. Sci. 270 (suppl_2) (2003) S186–S188.
[53] M.E. Newman, Modularity and community structure in networks, Proc. Nat. Acad. Sci. USA 103 (23) (2006) 8577–8582.
[54] P.M. Gleiser, L. Danon, Community structure in jazz, Adv. Complex Syst. 6 (04) (2003) 565–573.
[55] M. Höglund, A. Frigyesi, F. Mitelman, A gene fusion network in human neoplasia, Oncogene 25 (18) (2006) 2674–2678.
[56] S. He, G. Jia, Z. Zhu, D.A. Tennant, Q. Huang, K. Tang, J. Liu, M. Musolesi, J.K. Heath, X. Yao, Cooperative co-evolutionary module identification with application to cancer disease module discovery, IEEE Trans. Evol. Comput. 20 (6) (2016) 874–891.
[57] N. Zaki, J. Berengueres, D. Efimov, Prorank: a method for detecting protein complexes, in: Proc. ACM Int. Conf. Genetic Evol. Comput., 2012, pp. 209–216.
[58] H. Yu, P. Braun, M.A. Yıldırım, I. Lemmens, K. Venkatesan, J. Sahalie, T. Hirozane Kishikawa, F. Gebreab, N. Li, N. Simonis, et al., High-quality binary protein interaction map of the yeast interactome network, Science 322 (5898) (2008) 104–110.
[59] N. Binesh, M. Rezghi, Fuzzy clustering in community detection based on nonnegative matrix factorization with two novel evaluation criteria, Appl. Soft Comput. 69 (2018) 689–703.
[60] B. Rozemberczki, R. Sarkar, Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models, in: The 29th ACM Int. Conf. on Information and Knowledge Management, 2020, pp. 1325–1334.

Ronghua Shang (M'09) received the B.S. degree in information and computation science and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University in 2003 and 2008, respectively. She is currently a professor with Xidian University. Her current research interests include, evolutionary computation, image processing, and data mining.

Kejia Zhao received the B.E. degree in Electronic Information Science and Technology from Xi'an University of Architecture and Technology, Xi'an, China, in 2020. Now he is pursuing the M.S. degree in School of Artificial Intelligence, Xidian University, Xi'an, China. His current research interests include community detection and machine learning.

Weitong Zhang received the B.E. degree in Electronic and Information Engineering from Changchun University of Science and Technology, Changchun, China, in 2013, the M.S. degree in Electronics and Communication Engineering, and the Ph.D. degree in Electronic science and technology from Xidian University, Xi'an, China, in 2021. She is currently a lecturer with Xidian University. Her current research interests include complex networks, intelligent optimization, and machine learning.

Jie Feng (M'15) received the B.S. degree from Chang'an University, Xi'an, China, in 2008, and the Ph.D. degree from Xidian University, Xi'an, China, in 2014. She is currently an Associate Professor in the Laboratory of Intelligent Perception and Image Understanding, Xidian University, Xi'an, China. Her current interests include remote sensing image processing, deep learning, and machine learning.

Yangyang Li (SM'18) received the B.S. and M.S. degrees in computer science and technology, and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2001, 2004, and 2007, respectively. She is currently a Professor with the School of Artificial Intelligence, Xidian University. Her research interests include quantum-inspired evolutionary computation, artificial immune systems, and deep learning.

Licheng Jiao (F'17) received the B.S. degree in electronic engineering from Shanghai Jiaotong University, Shanghai, China, in 1982, the M.S. and Ph.D. degrees in electronic engineering from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. From 1990 to 1991, he was a Postdoctoral Fellow with the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, China. Since 1992, he has been a Professor with the School of Electronic Engineering, Xidian University. He is currently the Director with the Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University. He is in charge of about 40 important scientific research projects, and authored/coauthored more than 20 monographs and 100 papers in international journals and conferences. His research interests include image processing, natural computation, machine learning, and intelligent information processing. Dr. Jiao is a member of the IEEE Xi'an Section Execution Committee and the Chairman of awards and recognition committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, the Councilor of the Chinese Institute of Electronics, the Committee Member of the Chinese Committee of Neural Networks, and an expert of academic degrees committee of the state council.