Applied Soft Computing Journal 97 (2020) 106764

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Complex network graph embedding method based on shortest path and MOEA/D for community detection

Weitong Zhang *, Ronghua Shang, Licheng Jiao

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi Province, China

ARTICLE INFO

Article history:
Received 10 January 2020
Received in revised form 22 August 2020
Accepted 27 September 2020
Available online 2 October 2020

Keywords:
Graph embedding
Community detection
Shortest path
Decomposition multi-objective evolutionary algorithm

ABSTRACT

As one of the main applications of graph embedding, community detection has always been a hot issue in the field of complex network data mining. This paper presents a complex network graph embedding method based on the shortest path matrix and decomposition multi-objective evolutionary algorithm (SP-MOEA/D) for community detection, which can better reflect the network structure at the level of network community structure. Firstly, by calculating the shortest path matrix between nodes in the network, the node relationship matrix is obtained by adding the node similarity. Next, aiming at the problem of community detection in disconnected networks, a decomposition-based multi-objective optimization method is proposed to assign distances to unrelated nodes. Then, the network similarity matrix is calculated based on the relationship matrix of network nodes, and the low-dimensional vector representation of nodes is obtained by random surfing strategy and multi-dimensional scaling method. Finally, the community structure of the network can be detected based on the obtained node representation structure. Starting from the essence of network structure and the tightness between nodes, this method can reflect the relationship characteristics of network nodes more effectively, and then obtain the vector representation of nodes which can more accurately reflect the information of community structure in networks. The test results on 11 networks show that the node vector representation results obtained by this method can better reflect the community structure information in complex networks.

1. Introduction

Many complex systems can be represented by graphs, allowing efficient storage and access of relational knowledge about interactive entities [1]. Individuals in complex systems are abstracted as nodes in network graphs, and relationships among individuals are abstracted as edges. There are many forms of edges: weighted or un-weighted, directed or undirected, signed or unsigned, etc. [2]. According to the connection of nodes, the networks can also be divided into connected networks (there exists at least one path between any pairs of node) and disconnected networks (there is at least one pair of nodes with no path to reach) [3]. Graph not only reflects structured information, but also plays a key role in modern machine learning methods. Many machine learning methods use graph structured data as feature information to predict or discover new patterns [4]. Typical applications of graph analysis include: structure visualization, community detection [5,6], node classification [7], and connection prediction [8] etc. Community detection aims at finding sub-network structures with tight internal connections and sparse external connections. In addition to according to the observed connection relationships, the labels of nodes can be used to find the partition of the node set. In social networks, node labels may represent interests or beliefs; in citation networks, node labels may represent document topics or keywords; in biological networks, node labels may represent individual functions. The traditional community detection methods can be divided into two categories. One is to optimize the modularity [9] or modularity density [10] function. This kind of method is mainly based on multi-objective evolutionary optimization methods [11–13] and community integration methods. Due to the resolution limitation of modularity function and modularity density function, this kind of method may lose important network topology information although they can obtain good convergence results. The other one is based on network topology structure. For example, the methods based on spectral analysis [14] and the method based on random walk [15]. This kind of method mainly determines the similarity between nodes for community detection.

* Corresponding author.
E-mail addresses: zwt@stu.xidian.edu.cn (W. Zhang), rhshang@mail.xidian.edu.cn (R. Shang), lchjiao@mail.xidian.edu.cn (L. Jiao).

https://doi.org/10.1016/j.asoc.2020.106764
1568-4946/

Graph representation learning is a very extensive research direction in the field of graph mining in recent years. Each node in the graph structure is represented by a low-dimensional and dense vector, that is, the encoding process of nodes in the graph. This is called graph embedding or node embedding [16]. It is an important application direction of graph representation method to encode the high dimensional non Euclidean information of graph structure into feature vector and further discover community structure. Node embedding methods can be divided into three main categories: factorization based method, random walk based method and deep learning based method [17]. The algorithm based on factorization represents the connection between nodes in the form of matrix, and the matrix factorization is used to obtain the embedded representation of nodes. Locally Linear Embedding (LLE) method [18] assumes that each node is a linear combination of adjacent nodes in the embedded space. Laplacian Eigenmaps method [19] keeps two nodes embedded more closely when the weight of edges is higher. The method uses a quadratic penalty function for the distance between embeddings. Therefore, while maintaining node similarity, the difference information is often destroyed. Cauchy Graph Embedding [20] solves this problem by changing the formula of quadratic function. Hope method [21] uses Singular Value Decomposition (SVD) to obtain node similar matrices to maintain high-order approximation. Random walk models are often used to describe individuals moving in unpredictable ways [22]. Random walk method is especially useful when the graph size is too large to obtain the topological structure of the whole graph. In recent years, graph embedding method based on random walk model has been proposed and improved continuously. Deepwalk [23] is the earliest node embedding method based on random walk. Based on the node vector representation model of Word2vec [24], the random walk paths of nodes is constructed to imitate the process of text generation, and then obtain the sequence of nodes. Finally, Skip-gram and Hierarchical Softmax models are used to model the probabilities of the nodes in the sequence. Node2vec method [25] is further extended on the basis of Deepwalk. The random walk of the nodes in DeepWalk is a uniform and random distribution. Node2vec controls the breadth-first search and depth-first search in the random walk process by introducing two search bias parameters. Breadth-first search pays attention to the regional network representation, while depth-first search pays attention to the similarity between nodes. The objective functions of Deepwalk and Node2vec are non-convex and easily fall into local optimum [26]. HARP coarsens the graph by aggregating the nodes into the structure of the front layer, then generates the embedding of the coarsest graph, and improves the solution by better initialization of weight and avoids local optimum. Combining HARP with Deepwall and Node2vec based on random walk can get better result of graph embedding. FGE method [27] is a method based on open flow network model, which is used to reveal the underlying flow structure and hidden metric space of different random walk strategies on the network. It helps to find new potential applications in embedding. The above methods can reflect the network structure information well. But when further discovering the community structure of some non all connected networks, it is easy to appear fuzzy state at the edge of some communities. With more and more research on deep learning, a large number of deep neural network methods for graphics have been proposed. Structural Deep Network Embedding (SDNE) [28] uses a deep autoencoder to maintain the proximity of the first-order and second-order networks. The model consists of two parts: unsupervised and supervised. The former includes an autoencoder to find an embedded node that can reconstruct its neighborhood. The latter is based on Laplacian feature mapping. When similar nodes are far away from each other in the embedded space, the feature mapping will be punished. Inspired by PageRank [29], DNGR [30] combines random surfing with deep autoencoder to obtain network embedding results. The model consists of three parts: random surfing to obtain the sequence of nodes; generating probability co-occurrence matrix and transforming to positive pointwise mutual information (PPMI) matrix; superimposing denoising autoencoder. Graph Convolutional Networks (GCNs) [31] iteratively aggregates the neighborhood embedding of nodes, and uses the obtained embedding and its embedding function in the previous iteration to obtain new embedding. The aggregate embedding of local neighborhoods makes it scalable. Multiple iterations allow learning to embedding nodes to describe global neighborhoods. In addition to the above three typical node embedding methods, there are other methods, the more common is LINE method [32]. The method defines two functions for first and second order approximation respectively, and minimizes the combination of the two functions. The first-order adjacency function is similar to graph decomposition in order to keep the embedded adjacency matrix and dot product close. The difference is that LINE defines two joint probability distributions for each pair of nodes, one using adjacency matrix and the other using embedding.

Graph embedding method starts from the topological structure and node attributes of the graph, which is more helpful to understand the internal structure information and the relationship between nodes. Therefore, it has more advantages for further community detection applications. Due to the diversity of datasets in community detection, more focused methods should be proposed and studied. Some networks are not fully connected. In which some subgraphs or nodes are structurally independent. That is, there are no edges between them. When processing such data, the distance between some nodes is equal to infinity, and the similarity between them is 0. In addition, many datasets do not contain node attribute information. Therefore, it is more challenging to obtain the node vector representation results suitable for community detection problems with random walk method or graph information matrix based graph representation method. This paper will focus on the community detection of graph embedding method, which is an important application of network node embedding method. A decomposition-based multi-objective evolutionary optimization method is proposed to solve the problem of node embedding in disconnected networks. Firstly, the shortest path matrix between nodes in the network is calculated, and then the relational matrix between nodes is obtained by adding the similarity between nodes. The distance matrix between nodes can reflect the network topology and the tightness between nodes. As the name implies, the similarity function between nodes can reflect the degree of similarity between nodes. Combining the distance matrix and similarity between nodes, the initial sequence matrix of nodes reflecting the network structure can be obtained better. Aiming at the problem of community detection in disconnected networks, a decomposition-based multi-objective optimization method is proposed to assign distances to unrelated nodes. Two objective functions are designed, which are combined with the network core nodes (potential community structure center), so that the low-dimensional vector representation results of the nodes can better reflect the network structure and the tightness between nodes at the community structure level. Experiments show that the proposed method is more conducive to the application of node vector representation results to community detection in disconnected networks. After calculating the distance matrix and the similarity of nodes to get the network node relationship matrix, the network similarity matrix is further calculated, and then the low-dimensional vector representation of nodes is obtained by random surfing strategy and multi-dimensional scaling method.

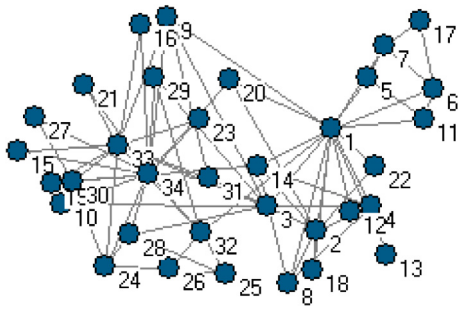


Fig. 1. A network with 34 nodes and 78edges.

Finally, the community structure of the network can be detected based on the obtained node representation structure. The main contributions of this paper are as follows:

(1) By combining the distance matrix with the similarity matrix, the initial sequence matrix of nodes reflecting the network structure can be obtained. It helps to collect the information contained in the network more comprehensively.

(2) A decomposition-based multi-objective optimization method is proposed to assign distances to disconnected nodes. It is helpful to improve the fuzzy community boundary caused by the infinite distance between some nodes in disconnected networks.

(3) Starting from the essence of network structure and the information of nodes, The proposed algorithm can reflect the relationship characteristics of network nodes more effectively, and then obtain more accurate expression of network node vectors which can better reflect the network structure at the level of network community structure.

2. Background

2.1. Graph representation for community detection

Given a graph G with n nodes and m edges. Graph embedding is a process of obtaining the node vector representation by random walk, matrix decomposition or deep learning. Community detection, one of the applications of graph embedding, is the process of dividing a graph into several subgraphs according to its topological structure and node attributes. The internal nodes of these subgraphs are closely connected, while the external links are sparse. The graph embedding method maps the topological structure and node attributes of the graph to the node vector representation. For the graph structure shown in Fig. 1, the visualization result of the node vector representation obtained by graph embedding method is shown in Fig. 2(a). Then using the node vector in Fig. 2(a), a simple clustering method can be used to further obtain the community detection results, as shown in Fig. 2(b). The nodes in the rectangular box belong to the same community. This paper is a further study of community detection based on graph embedding.

2.2. Random walk based node embedding

Since the vectorization model of DeepWalk nodes based on Word2vec was proposed, the network embedding method based on random walk has been proposed continuously. These methods generate the sequence of nodes by random walk strategy, and finally obtain the result of node embedding. Compared with DeepWalk, Node2vec introduces breadth-first search and depth-first search into the generation of random walk sequences by introducing two parameters p and q. Probability p controls the probability of jumping up to the neighbor of the last node, and q controls the probability of jumping up to the non-neighbor of the last node. Given the source node$i$, and simulating a random walk with length l. Let c_h be the hth walking node and the starting node c_0 is set to i. Node c_h follows the following distribution:

$$P(c_h = j | c_{h-1} = i) = \begin{cases} \frac{p_{ij}}{Z} & if\ (i,j) \in E \\ 0 & otherwise \end{cases} \quad (1)$$

where p_{ij} is the transition probability from node i to node j, Z is used for standardization.

It can be seen that the method of obtaining sampling sequence of nodes by random walk needs to set the sequence length in advance. The neighbor information of fringe nodes is often difficult to be captured comprehensively. In addition, it is difficult to determine parameters such as step size and step number of random walk. The random walk strategy reflects the neighborhood information of nodes, the sequence information of unrelated nodes will be very different when dealing with disconnected networks. The node vectors obtained in this way will have some influence when they are applied to community detection. This paper calculates the shortest path matrix of network nodes, and then calculates the similarity matrix of network nodes according to the shortest path matrix and the similarity between nodes.

2.3. Floyd–Warshall

Floyd–Warshall algorithm [33] is a common algorithm for finding the shortest path in graph in dynamic programming. The pseudocode of the Floyd–Warshall algorithm is as follows:

Algorithm 1: Floyd-Warshall algorithm

```
Input: Network Adjacency Matrix A;
dist=A;
for k=1: n do // k is the "intermediate node"
    for i=1: n do
        for j=1: n do
            if (dist(i, k) + dist(k, j) < dist(i, j)) then
                dist(i, j)=dist(i, k) + dist(k, j);
Output dist.
```

where n is the number of nodes, i, j are any nodes in the network. It can be seen that the shortest path matrix between nodes can reflect the tightness between nodes from the reverse side. The shortest path between nodes is larger, which indicates that the two nodes are less closely connected.

2.4. Random surfing

Random surfing is mainly inspired by PageRank. PageRank, also known as web page ranking, is a technology used by search engines to calculate web page ranking based on hyperlinks between web pages. In Random surfing model, the transition matrix T is defined to represent the transition probability between different nodes. Introducing row vectors p_k, The jth entry denotes the probability of reaching node j after k-step transfer. p_0 is a vector with only one term of 1 and all the others are 0. Random surfing considers the case of restart, that is, there is a certain probability to return to the initial node. There are the following relationships:

$$p_k = \alpha \cdot p_{k-1} T + (1-\alpha) p_0 \quad (2)$$

The shortest path matrix of network nodes is calculated first. Then the network node similarity matrix based on the shortest distance matrix is calculated. The node similarity matrix is used as the transition matrix in Random Surfing to obtain the sequence of nodes.

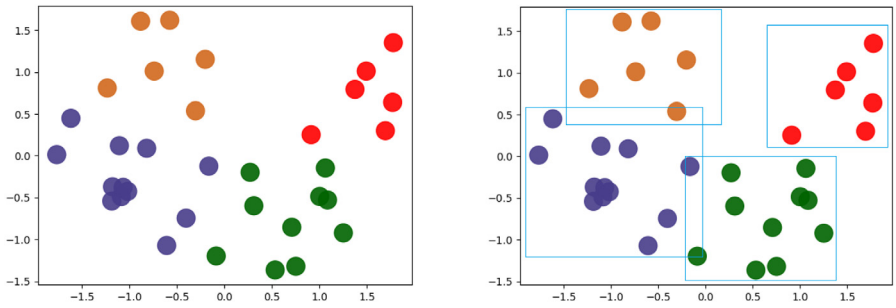


Fig. 2. Visualization results of the network in Fig. 1. (a) Node vector representation result. (b) Community detection result.

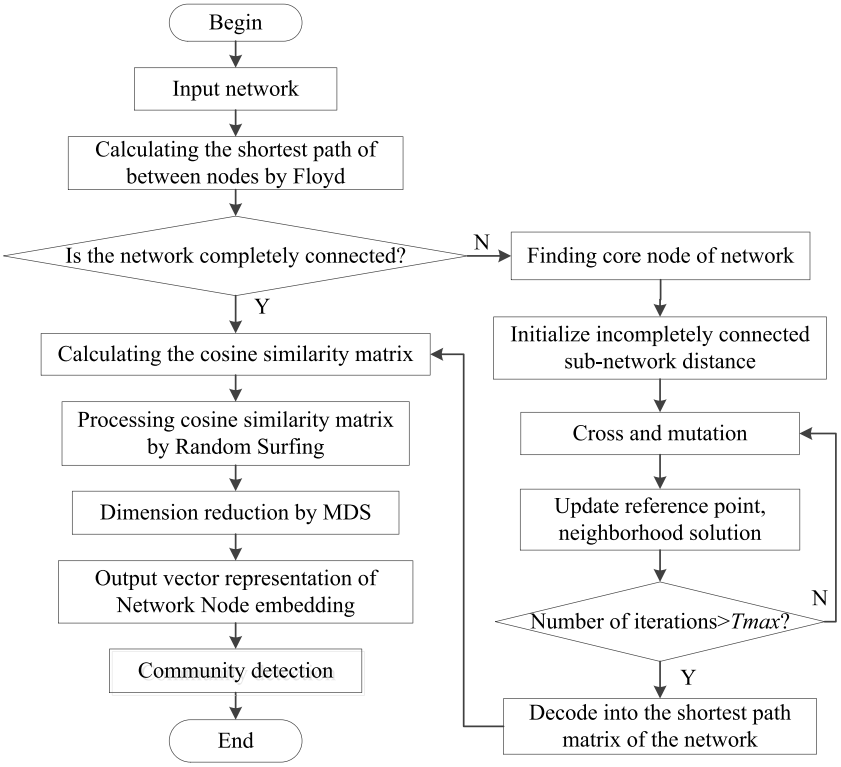


Fig. 3. The flow chart of the proposed algorithm SP-MOEA/D.

3. Graph representation method for complex network

A network node embedding method based on the shortest path and similarity matrix of nodes is proposed for the important application of graph embedding method: community detection. A decomposition-based multi-objective evolutionary optimization method is proposed to solve the problem of node embedding in disconnected networks. Firstly, the shortest path matrix between nodes in the network is calculated, and the node similarity is added to obtain the network node relationship matrix. To solve the problem of community detection in disconnected networks, a decomposition-based multi-objective optimization method is proposed to assign distances to unrelated nodes. Then, the network similarity matrix is calculated based on the relationship matrix of network nodes. Finally, the low-dimensional vector representation of nodes is obtained by random surfing strategy and multi-dimensional scaling method. Finally, the community structure of the network can be detected based on the obtained node representation structure. In summary, the flow chart of the proposed algorithm SP-MOEA/D is shown in Fig. 3.

3.1. Similarity matrix based on shortest path and similarity between nodes

A method for obtaining the vector representation of network nodes by Multidimensional Scaling (MDS) [34] is proposed. Before adopting the MDS method, the initial representation of the network node is needed. Community detection is the discovery of a set of nodes with tight internal connections and sparse external connections in complex networks. Therefore, more attention should be paid to the network topology information and the connections between nodes for the representation of node vectors in community detection. Floyd–Warshall algorithm is used to calculate the shortest path matrix between nodes in the network. The elements in the matrix correspond to the minimum number of edges that need to be passed for each pair of nodes to form a connection. Thus, the matrix reflecting the network structure and the tightness between nodes can be obtained preliminarily. Then, the similarity between nodes in the network is calculated and added to the shortest path matrix after negative to obtain a new

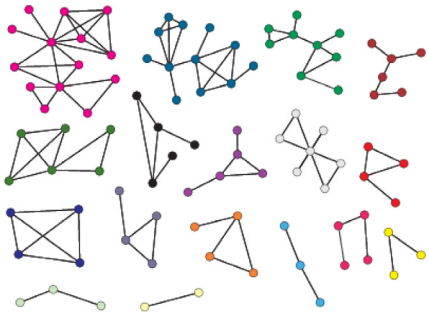


Fig. 4. Part of the sub-network in the Netscience network.

matrix. The cosine similarity function [35] is used to calculate the similarity between nodes. The cosine similarity function is expressed as follows.

$$Sim = \frac{U \cdot V}{\|U\| \, \|V\|} \tag{3}$$

where U and I represent any two vectors. For the similarity between nodes in complex networks, the cosine similarity function can be rewritten as follows:

$$Sim = \frac{\left|N_i \cap N_j\right|}{\sqrt{|N_i| \, \left|N_j\right|}} \tag{4}$$

where i and j represent any node in the network, $N_i(N_j)$ represents neighbor sets of node i (j).

After getting the shortest path between nodes and the similarity matrix between nodes, the relationship matrix between nodes is obtained after adding the two. The elements of each row in the relationship matrix are regarded as the characteristics of each node in the network, which indicates the tightness between this node and other nodes. Then the similarity matrix is further calculated for the relationship matrix between nodes. When Floyd–Warshall is used to calculate the shortest path matrix among network nodes, there will be a situation where the distance between nodes is infinite in disconnected networks. In Section 3.3, a method obtaining a distance matrix between nodes in a network is presented.

3.2. Connected sub-networks and core nodes in disconnected network

In an disconnected network, there must be some fully connected sub-networks. For example, the Netscience network [36] is a disconnected network, which consists of 1589 nodes and 2742 edges. In Netscience network, there are 395 completely independent sub-networks, the maximum node degree is 34, the minimum node degree is 0, that is, there are nodes with no neighbor nodes in the network. Part of the sub-network in the Netscience network is shown in Fig. 4.

When the decomposition based multi-objective evolutionary algorithm is used to obtain the shortest path matrix among nodes in an disconnected network, the value of one of the objective functions is determined by the network core nodes. All nodes in the network with the largest degree in the neighborhood are taken as the core nodes. The core nodes in the sub-network or network community structure are often more closely connected with other nodes in the network [37]. A node belongs to the "destructive decisive critical" core node when the node is removed or changed to affect the structure or robustness of the whole network. Another kind of core node, which is obtained by analyzing the information such as the centrality index of nodes in the network, belongs to the "significant equivalent to critical" core node. Among them, the calculation indexes of node centrality are mainly as follows: centrality index based on edge betweenness index, centrality index based on compactness degree and centrality index based on node degree.

3.3. Graph embedding method based on MOEA/D for disconnected network

The network similarity matrix is calculated by cosine similarity function and distance between network nodes. The initial sequence of network nodes is obtained. Then, the low-dimensional vector representation of network nodes is obtained by random surfing and multi-dimensional scaling method. For disconnected networks, the sequence information of unconnected nodes will be very different. Therefore, this paper introduces a multi-objective evolutionary optimization method to obtain the distance matrix between nodes in the network, which can reflect the similarity degree and attribute characteristics of nodes at the same time. Multi-objective evolutionary optimization algorithm has been applied to many fields because it can obtain a set of Pareto optimal solutions with different emphasis [38,39]. This section mainly introduces the multi-objective optimization algorithm based on decomposition, which assigns the shortest path to the unconnected nodes in the disconnected network.

3.3.1. Objective functions

In order to obtain a more shortest path matrix of nodes, two objective functions are designed as follows.

$$\begin{cases} \min f_1 = \sum\limits_{a=1,b=a+1}^{r} d_{ab} \\ \min f_2 = -std(\sum\limits_{l=1}^{k-1} d_l) \end{cases} \tag{5}$$

where r represents the number of sub-networks, d_{ab} represents the allocation distance between sub-networks in network, a and b represent any two sub-networks, std represents the standard deviation operation, d_l represents the distance between any core node and other core nodes described in Section 3.2. In the MOEA/D-Net framework, each sub-problem is represented as follows.

$$\begin{aligned} &\min g_i(x|\lambda_i, z^*) = \max\{\lambda_{i1}|f_1(x) - z_1^*|, \lambda_{i2}|f_2(x) - z_2^*\} \\ &Subject \quad to \quad x \in \Phi \subseteq \mathrm{R}^{\mathrm{N}} \end{aligned} \tag{6}$$

where $\lambda_i = \{\lambda_{i1}, \lambda_{i2}\}$ represents weight coefficients, $z^* = (z_1^*, z_2^*)$ represents a reference point, where

$$\begin{aligned} &\lambda_{i1} = (i-1)/(N_p - 1); \lambda_{i2} = (N_p - i)/(N_p - 1) \\ &z_1^* = \{\min f_1(x)|x \in \Phi\}; z_2^* = \{\min f_2(x)|x \in \Phi\} \end{aligned} \tag{7}$$

For each non-dominant point x*, there should be an appropriate weight vector λ to make x* the optimal solution of the above formula. The non-dominant solution set can be obtained by simultaneously optimizing the sub-problems under these different weight vector values λ.

Both objective functions are minimized at the same time. The first objective function calculates the mean value of the distribution distance between all unrelated sub-networks, that is, the mean value of all genes in the chromosome. The second objective function calculates the negative standard deviation of the distance between any core node and other core nodes described in Section 3.2. In the follow-up experiments of this paper, the first core node is chosen as the arbitrary core node.

The motivation for the design of these two objective functions is as follows. This paper focuses on the important application

of community detection based on graph embedding method, so it focuses on the close relationship between nodes and network structure characteristics. When distributing the distances of disconnected nodes, the integrity of the original network topology information should be ensured, and the distribution results should be guaranteed to be conducive to further community structure detection. When the node allocation distance is initialized, the initialization range should be larger than the maximum node degree of the sub-network where the node is located, and less than the maximum node degree plus the number of unrelated sub-networks in the network. When the scale of the network becomes larger, the number of disconnected sub-networks will also increase. Therefore, in order to better ensure the distribution result is conducive to further community structure detection, the first objective function is to control the distribution distance of nodes so that it will not be too large. In addition, there should be a restrictive relationship between the multiple objectives in the multi-objective optimization method, so as to control the solution obtained by the algorithm to evolve towards a more advantageous direction. The core node in the network can be regarded as the potential community center node in the network, so the distance between the core nodes can reflect the distance between the potential community structures in the network. The second objective function is to widen the distance between the core nodes in the network. As you can see from the previous introduction, there will be at least one core node in each unconnected subnet. Increase the distance between the core nodes and ensure that the node allocation distance is not too large. In this way, the result of node vector representation is more conducive to community structure detection in the network. In the following experiments, the validity of the two objective functions will be further proved.

3.3.2. Algorithm framework

The multi-objective optimization method can find a series of effective solutions in an independent operation, and the obtained series of solutions always have diversity [40,41]. Among them, MOEA/D is one of the widely studied methods in recent years [42]. The MOEA/D has the following advantages. The framework of MOEA/D-Net algorithm is adopted to solve the distance assignment tasks of disconnected nodes in disconnected networks, so as to obtain a series of more advantageous solutions. The algorithm framework is as follows:

Firstly, a framework based on the decomposition is established. The framework decomposes complex multi-objective optimization problems into several simple sub-problems of single-objective optimization, which can solve multi-objective optimization problems more efficiently. Secondly, the decomposition-based optimization method can solve several sub-problems simultaneously through the evolution of several chromosomes (solutions), so that the algorithm can run independently at each time. In addition, the decomposition-based optimization method combines the neighborhood information of each sub-problem to optimize them. This optimization method can quickly spread useful information containing specific structural knowledge to the sub-problems to be solved and their neighborhood problems. The computational cost is low. The multi-objective optimization problem based on decomposition adopts implicit strategies of uniformly distributed sub-problems to ensure the diversity of solutions. After obtaining a series of Pareto optimal solutions with different emphasis on the objective functions, these solutions are decoded as the distance matrix between nodes in the network. The node vector representation results are obtained by further processing. Due to the different emphasis of the solution set on the objective functions, it is more helpful to obtain the results of different evaluation indexes.

3.3.3. Initialization

Firstly, the connected sub-network structure in the network is searched and the maximum node degree of each sub-network node is calculated. Then, according to the maximum node degree of the sub-network, initialization assignment is made for the distance between the nodes of the sub-network. The structure of Karate network [43] is modified and introduced as a disconnected network. The specific method is to separate some nodes in Karate network and get a new network Karate_t. Fig. 5 shows the original Karate network and variant Karate_t.

Fig. 5(a) is the structure of the original Karate network, which has 34 nodes and 78 edges. It is a connected network. The original Karate network contains two community structures, which are distinguished by circles and triangles. Fig. 5(b) is a variant of the Karate network, Karate_t. There are three disconnected sub-networks. The new independent sub-networks contain nodes 7, 17, 26, and 9, 12, 31, 32. The remaining nodes and edges remain unchanged. In addition to the two community structures in the original network, the two independent sub-networks will be two new community structures. The communities are distinguished using circles, triangles, diamonds and squares in Fig. 5(b).

As shown in Fig. 5(b), the connected sub-networks in Karate_t network are searched and the maximum node degrees in each sub-network are calculated, which are 14, 2 and 2, respectively. Since the distance allocated to nodes between sub-networks is infinite in the original distance matrix, the distance allocated to nodes should not be less than the distance between nodes and neighboring nodes in order to ensure that the topology and close relationship of the original network are not destroyed. In addition, if a distance value is reallocated for each pair of unrelated nodes, the complexity of the algorithm will be very high and the convergence speed of the algorithm will be greatly affected. Therefore, the form of the solution is as follows:

$$X = \{x_1, x_2, \ldots, x_{N_p}\} \tag{8}$$

where x_i represents the gene value in chromosome X, N_p represents the number of genes, $N_p = k(k-1)/2$, k represents the number of sub-networks in the network.

As shown in Fig. 5(b), the Karate_t network consists of three sub-networks, $k = 3$, $N_p = 3$. Therefore, the solution of Karate_t network is $\{x_1, x_2, x_3\}$. Where x_1 represents the allocation distance between all nodes in sub-network 1 and all nodes in sub-network 2, x_2 represents the allocation distance between all nodes in sub-network 1 and all nodes in sub-network 3, x_3 represents the allocation distance between all nodes in sub-network 2 and all nodes in sub-network 3. The distances of all nodes in each group of sub-networks are equal. The distance is initialized as:

$$d_{ab} = \max(y_a, y_b) + rand(k) \tag{9}$$

where y_a represents the maximum degree of nodes in sub-network a, rand(k) represents the operation of randomly generating integers between 0 and k.

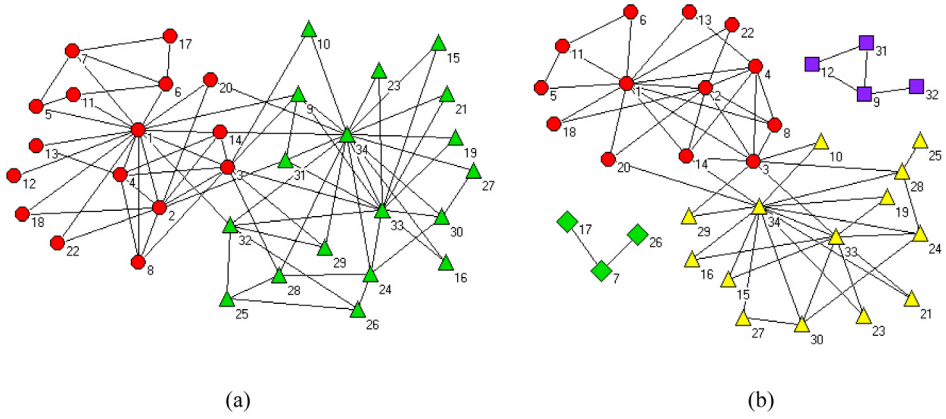
This allocation method saves a lot of computational complexity, and the distance between sub-networks is not less than the distance between nodes in sub-networks, which ensures the original network topology.

3.3.4. Genetic operators

The genetic operators mainly include crossover operator, mutation operator and selection operator. In the process of chromosome optimization, through the operation of chromosome replication, crossover or mutation, the population evolves continuously, and converges to the "most suitable environment" population. A series of effective non-dominant solutions are obtained. In the process of population evolution, the crossover operator mainly ensures that the offspring inherit the characteristics of the

Algorithm 2: The framework of MOEA/D-Net

Input: N_p sub-problems, crossover probability p_c, mutation probability p_m, neighborhood size N_s, updating scale: N_u, Maximum number of iterations g_{max};
for t_1=1: g_{max} do
  Initialize population: $X=\{x_1, x_2, \ldots, x_{Np}\}$;
  $z^*=10^6$; // z^* is Reference Point
  for t_2=1: N_p do
    x'←Crossover operation to x; // x' is offspring solution
    x''←Mutation operation to x'; // x'' is offspring solution
    x_u←x'' (with minimum g_i); // x_u is the renewal solution
    for t_3=1:j // j=1, 2, ..., Np
      if $z_j^*>f_j(x_u)$
        $z_j^*=f_j(x_u)$;
    for t_4=1:N_u
      if $g_j(x_j \mid \lambda_i, z^*)>g_j(x_u \mid \lambda_i, z^*)$
        x_j←x_u; // x_j is neighborhood solutions of the sub-problem g_j
Output: X_s. // The compromise series solution set



(a) (b)

Fig. 5. Karate network and its variant structure. (a) The original Karate network structure. (b) Karate_t: the variant structure of Karate Network.

paternal individuals. The mutation operator obtains new offspring individuals mainly by random perturbation of chromosome itself.

Two-point crossover is used as the crossover operator in the multi-objective optimization method. Specific operations are as follows: given two parents x_i and x_j ; randomly select two points a and b ($1 \leq a \leq b \leq r$); generate a random number between (0, 1); if the random number is less than the crossover probability p_c, the gene values of two paternal chromosomes a and b are exchanged.

The mutation operator adopts the single point mutation operator. The mutation is only for chromosomes obtained by the crossover operator introduced earlier. The specific operations are as follows: for each gene on the chromosome obtained by the crossover operator, a random number between (0,1) is generated; if the random number is less than the mutation probability p_m, change the value in this position. The specific changed value is the same as that the generation of d_{ab}. Such variation methods and values will ensure the diversity of individuals in the evolutionary process and help to further search for effective non-dominant solutions.

3.4. Dimensionality reduction method based on multi-dimensional scaling method

According to the operation in Sections 3.1–3.3, the network node relationship matrix is obtained firstly by obtaining the shortest path and node similarity between network nodes. Then the elements in the matrix are used as network node characteristics for further similarity calculation. The network similarity matrix is processed by random surfing strategy. Finally, the low-dimensional vector representation of network nodes is obtained by dimensionality reduction by multi-dimensional scale scaling method. The basic idea of multi-dimensional scaling method is to map points in high-dimensional coordinates into low-dimensional space, while keeping the distance (similarity) between nodes as constant as possible. The concrete operation steps are as follows: firstly, the distance matrix of the nodes in the original space is calculated; the inner product matrix is calculated according to the distance matrix; secondly, the eigenvalue matrix and eigenvector matrix are obtained by eigenvalue decomposition of the inner product matrix; finally, the first M term in the eigenvalue matrix and its corresponding eigenvector are remained, where M is the final vector dimension. The distance matrix of the nodes in the original space is the network similarity matrix. Similarity matrix can be regarded as a matrix reflecting the distance between points in the original space.

3.5. Complexity analysis

Assuming there are n nodes and m edges in the graph. For connected networks, the complexity of computing the shortest

Table 1
Connected network information.

Network	Node	Edge
Karate	34	78
Dolphin	62	159
Football	115	613
Polbooks	105	441
SFI	118	200
Power	4941	6594

Table 2
Disconnected network information.

Network	Node	Edge
Karate_t	34	62
Dolphin_t	62	136
Email-Eu-core	1005	16064
Polblogs	1490	16717
Netscience	1589	2742

distance matrix, computing node similarity, random surfing and MDS is $O(n^2)$, and the complexity of K-means is $O(n)$. Therefore, the time complexity of the proposed algorithm for connected networks is $4O(n^2)+O(n)\approx O(n^2)$. For disconnected networks, the complexity of finding the core node is $O(\mathrm{m})$. In MOEA/D algorithm, The operation of crossover, mutation operation, selecting updating solution and updating neighborhood solution need $O(n)$ basic operations, and the time complexity of updating reference point is $O(1)$. Therefore, the time complexity of the proposed algorithm for disconnected networks is $4O(n^2)+O(n)+4O(n)+O(m)+O(1)\approx O(n^2)$.

4. Experimental results and analysis

4.1. Comparison algorithms and datasets

In order to better analyze the effectiveness of this algorithm, this section compares the algorithm with the following methods: Deepwalk method, Node2vec method, LINE method, DNGR method, DNE-SBP method [44] and ECD method [45]. The ECD method uses evolutionary computation method to detect community directly. There is no graph embedding results, so the visualization results of the ECD method are not compared.

After obtaining the node vector representation by the above six graph embedding methods, the community structure of the node representation results is detected. In this paper, K-means clustering method is used. In addition, the network similarity matrix obtained in this algorithm is used as the data matrix input in the DNGR algorithm. After obtaining the node vector representation through the above 6 methods, the community structure of the node representation results can be detected. The K-means clustering method is used in this paper. In this experiment, it is applied to unsigned networks for testing. The information of 6 connected and 5 disconnected real-world networks are given in Tables 1 and 2.

Karate network is a network built according to the community relationship among members of a Karate club. The Karate network consists of 34 nodes and is divided into two communities due to differences between club directors and coaches. The Dolphin network was acquired by Lusseau and others when they studied a group of bottlenose dolphins living in Magic Bay, New Zealand. The network consists of 62 nodes. The network is divided into two communities due to the separation of dolphins labeled "SN100" for a certain period of time. Football network is the competition network of American college football team in autumn 1999 regular season. It contains 115 nodes representing 115 football teams from 12 leagues. The nodes in Polbooks Network [12] represent books on American politics sold by amazon.com, an online bookstore. The SFI network represents 271 scientists and their collaborators staying underground at the Santa Fe Institute and other institutions at any time from 1999 to 2000. The edges of the network represent two scientists working together on one or more projects at the same time. Power network is an undirected network, which represents the topology of the western power grid in the United States. There are 4941 nodes representing 4941 power base stations.

In order to more intuitively compare the graph representation results of various algorithms on the disconnected network, the Karate network and the dolphin network are transformed into the Karate_t network and the dolphin_t network. Karate_t network separates several nodes in Karate network to form two sub-networks. The independent nodes are node 7, 17, 26, and node 9, 12, 31, 32. The remaining nodes and edges remain unchanged. Similarly, the Dolphin_t network separates several nodes in the Dolphin network into two sub-networks. The independent nodes are node 5, 12, 29, 40, 52, and node 13, 22, 34. The remaining nodes and edges remain unchanged. The Email-Eu-core network [46] is generated from e-mail data from a large European research institute. E-mail represents the communication between members of an organization. When there is communication, an edge is generated. The network contains 42 community structures. Polblogs Network [47] is a data set of political blogs. The connections between blogs in the network are automatically extracted from the front page of the blog. Netscience network is a compilation of review bibliographies on two networks. 1589 nodes represent 1589 co-authors.

4.2. Evaluation indexes

There are three main evaluation indicators for community detection: Normalized Mutual Information (*NMI*) [48], Modularity (*Q*), Modularity Density (D).

(1) Normalized Mutual Information (NMI)

Normalized Mutual Information detects the validity of network partitioning results based on real network partitioning. For two different partitions *A* and *B*, *NMI* is defined as follows:

$$NMI=\frac{-2\sum_{u=1}^{C_A}\sum_{v=1}^{C_B}C_{uv}\cdot\left(\frac{C_{uv}\cdot n}{C_u\cdot C_v}\right)}{\sum_{u=1}^{C_A}C_u\log\left(\frac{C_u}{n}\right)+\sum_{v=1}^{C_B}C_v\log\left(\frac{C_v}{n}\right)} \tag{10}$$

where n is the number of nodes, C is confusion matrix, the element C_{uv} in the matrix denotes the number of nodes belonging to u community in A partition and v community in B partition. $C_A(C_B)$ is the number of communities in $A(B)$ partition, $C_u.(C_v)$ is the sum of the elements in line $u(v)$ of matrix C. The greater the value of *NMI*, the more similar the division A and B are. When the value of *NMI* is 1, the division A and B are identical.

(2) Modularity (Q)

Modularity function is the most commonly used indexes to evaluate the structure of network community. The greater the modularity, the closer the internal connection of the community structure in the network, that is, the better the result of network partition. For an un-weightless and undirected network, the modularity function is as follows:

$$Q=\frac{1}{2m}\sum_{c=1}^{N}\left[2l_c-\frac{(d_c)^2}{2m}\right] \tag{11}$$

where N is the number of communities, m is the total number of edges in the network, c is the community number, l_c is the total number of edges within the community C, d_c is the sum of the node degree of the nodes in the community C.

(3) Modularity Density (D)

For the resolution of modularity, another commonly used evaluation criterion in community detection is modularity density function. The modularity density function is expressed as follows:

$$D = \sum_{u=1}^{N} \frac{L(c_u, c_u) - L(c_u, \overline{c_u})}{|c_u|} \tag{12}$$

where c_u is a community in the network partition result, $L(c_u, c_u)$ is the number of connections between the nodes in the c_u community, $L(c_u, \overline{c_u})$ is the number of connections between nodes within the community c_u and the nodes outside the c_u community, $|c_u|$ is the number of nodes in community c_u.

4.3. Parameter analysis

In this section, the parameter selection in this algorithm will be explained experimentally. The main parameters are: probability parameterα in Random Surfing strategy, population size *popsize*, maximum iteration number g_{max}, sub-problem neighborhood size N_s, crossover probability p_c, mutation probability p_m.

The parameter α in Random Surfing strategy represents the probability that the operation will continue in the random surfing process. 1-α represents the probability that the surfing process will return to its original node and restart. Keep the other parameters in the algorithm unchanged, Fig. 6 shows the effect of α parameters in [0.9, 1] interval on the experimental results on four networks. *popsize* is set to 100, g_{max} is set to 30, and N_s is set to 15.

As can be seen from Fig. 6, when α is taken in the range of [0.9, 0.98], the impact on the results of community detection on four networks is not obvious, but when the value is higher than 0.98, the results decrease significantly. Therefore, the parameter α is taken in the interval of [0.9, 0.98] according to the different network.

Fig. 7 shows the effect of parameter *popsize* on experimental results on three networks. Among them, α is set to 0.98, g_{max} is set to 30, and N_s is set to 15. The values of the three evaluation indexes on Email-Eu-core network are quite different. For the convenience of display, double ordinates are adopted in Email-Eu-core network. The right ordinate corresponds to the D value. So as Figs. 8–9. In evolutionary algorithms, too small population size can easily lead to poor individual diversity of the generated offspring. When the population size is too large, redundant individuals are prone to occur. Therefore, the population size is chosen between [60, 200].

As can be seen from Fig. 7, the size of the population has little effect on the results of small-scale networks. For large-scale networks, too small population size settings may lead to poor iteration results. Therefore, the *popsize* parameter is set to 100.

Set α to 0.98, *popsize* to 100 and N_s to 15. Test the effect of g_{max} parameters on experimental results on three networks as shown in Fig. 8. As can be seen from Fig. 8, the setting of the maximum number of iterations has little effect on the results of small-scale networks. For large-scale networks, too small the maximum number of iterations may lead to poor iteration results. Therefore, the maximum number of iterations g_{max} parameter is set to 30.

The effect of the values of N_s parameters on the experimental results is tested on three networks as shown in Fig. 9. The α is set to 0.98, *popsize* is set to 100, g_{max} is set to 30. As can be seen from Fig. 9, the setting of sub-problem neighborhood size has little effect on the results of small-scale networks. For large-scale networks, too large or too small neighborhood size settings of sub-problems may lead to poor iteration results. Therefore, the value of the neighborhood size N_s of the sub-problem is set to 15.

p_c and p_m are the crossover probability and mutation probability of MOEA/D algorithm. Based on the experience of evolutionary algorithm, the values of p_c and p_m are 0.8 and 0.5, respectively.

4.4. Visualization results

In this section, several visualization results of node embedding on the network are given for the algorithm and the comparison algorithm. The nodes with different shapes and colors represent the distribution of nodes in the real network partition results. Fig. 10 shows the visualization results of six algorithms in Karate network. The horizontal and vertical coordinates represent the coordinates of node vector representation results. Therefore, the relationship and distribution of the nodes in the embedded results are more clear. So as Figs. 11–13.

As can be seen from Fig. 10, the visualization results of Deepwalk algorithm and Node2vec algorithm in the node embedding problem are very similar and the effect is very good. The advantages of LINE method and DNE-SBP algorithm in visual results are not obvious. In the case of node embedding obtained by DNGR algorithm, the location of a few nodes is inaccurate. As can be seen from the visualization result obtained by the proposed algorithm, the location of nodes in the same community is relatively centralized, which can well reflect the node relationship in the network.

Fig. 11 shows the visualization results of six algorithms node embedding in Dolphin network.

As shown in Fig. 11, the results of Dolphin network embedding obtained by Deepwalk and Node2vec algorithms are still similar, and they can well reflect the tightness between nodes in the network. Both LINE algorithm and DNGR algorithm have a few nodes whose locations cannot effectively reflect the tightness between nodes in the network. As can be seen from the visualization result obtained by the proposed algorithm, although the overall network node distribution is relatively compact, it can still reflect the relationship between nodes in the network through node embedding.

Fig. 12 is the visualization results of six algorithms embedding nodes in Karate_t network. From Fig. 12, it can be seen that the boundary between the independent sub network and other nodes is not clear in the embedded results of the Karate_t network nodes obtained by Deepwalk algorithm, Node2vec algorithm and DNGR algorithm. Results of LINE algorithm and DNE-SBP do not reflect the relationship between independent sub-network and other nodes very well. However, the proposed algorithm can still accurately describe the tightness between nodes in the network.

Fig. 13 shows the visualization results of six algorithms embedding nodes on Dolphin_t network. As shown in Fig. 13, the Deepwalk and Node2vec algorithms can identify the independent sub-networks in the network more accurately. However, the algorithm based on random walk is affected by the large distance between independent sub-networks, which can easily blur the boundaries of nodes between connected communities. This will lead to inaccurate application in community detection. The results of LINE algorithm and DNGR algorithm are poor when dealing with complex networks with independent sub-networks.

It can be seen that the proposed algorithm can identify independent sub-networks accurately, and effectively distinguish the original connected communities, which is more conducive to the application of complex network embedding results to community detection tasks.

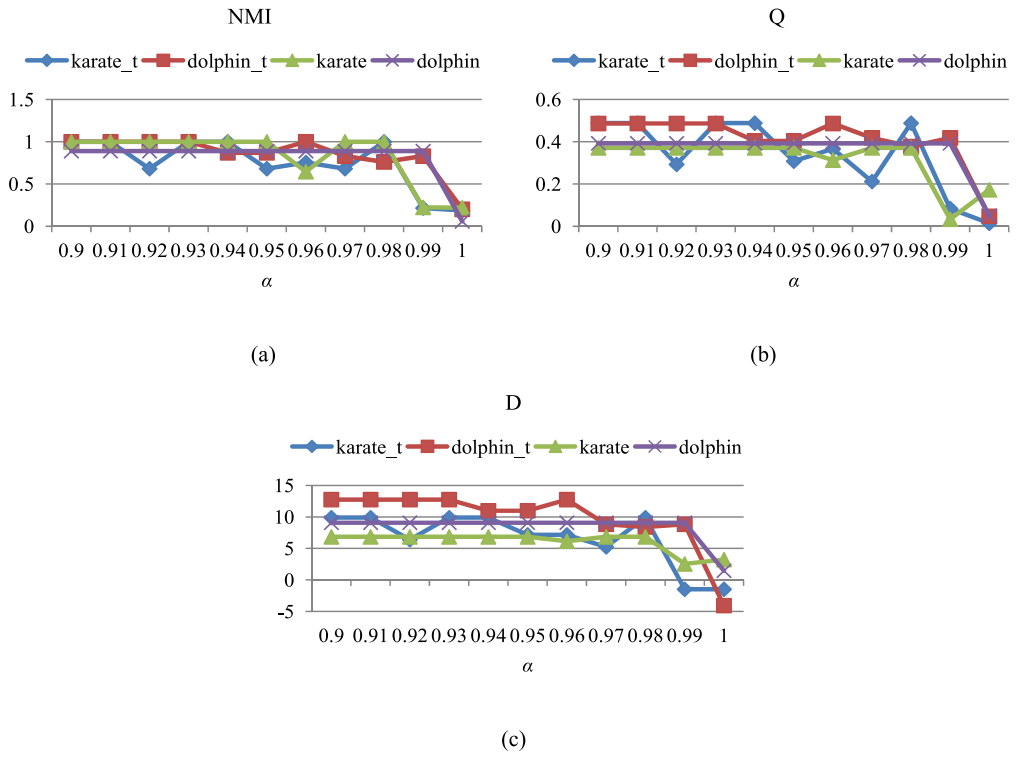


Fig. 6. The effect on three indices of parameter α values on four networks. (a) *NMI*. (b) *Q*. (c) *D*.

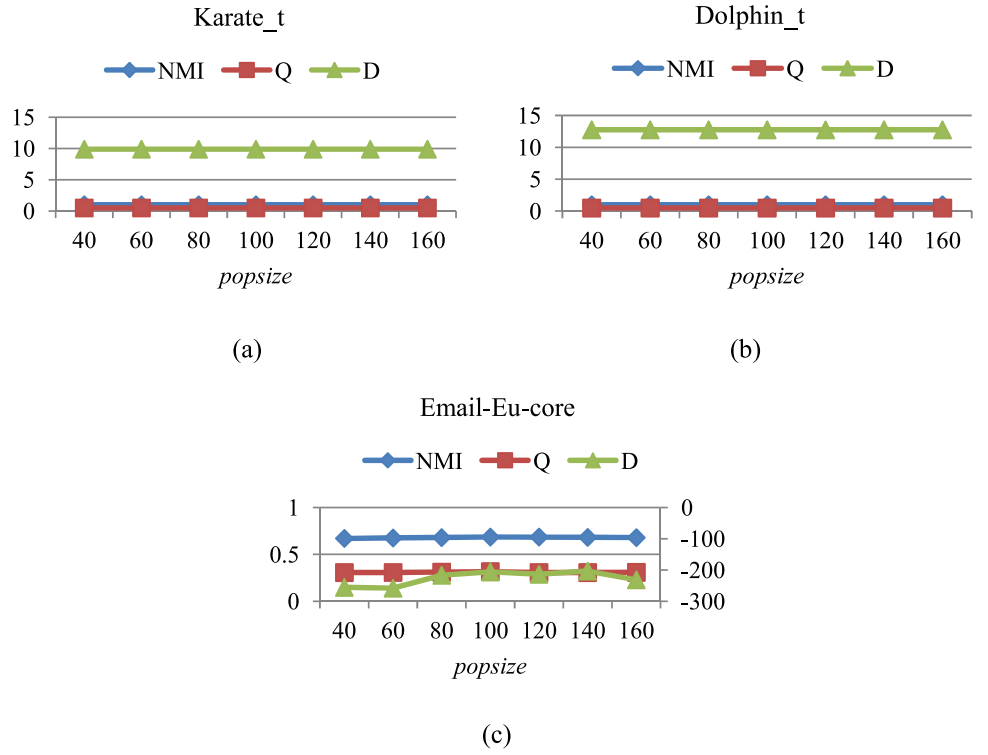


Fig. 7. The effect of parameter *popsize* on three networks. (a) Karate_t. (b) Dolphin_t. (c) Email_Eu_core.

4.5. Community detection results

In order to further verify the effectiveness of the proposed algorithm, this section will give the results of community detection of network embedding in connected and in connected networks. Table 3 shows the *NMI* values obtained by six algorithms on eight networks with real partitions.

From Table 3, we can see that the network embedding results obtained by six embedding methods are applied to eight kinds of network community detection with real partition. Deepwalk algorithm obtains the max *NMI* values in Dolphin network and

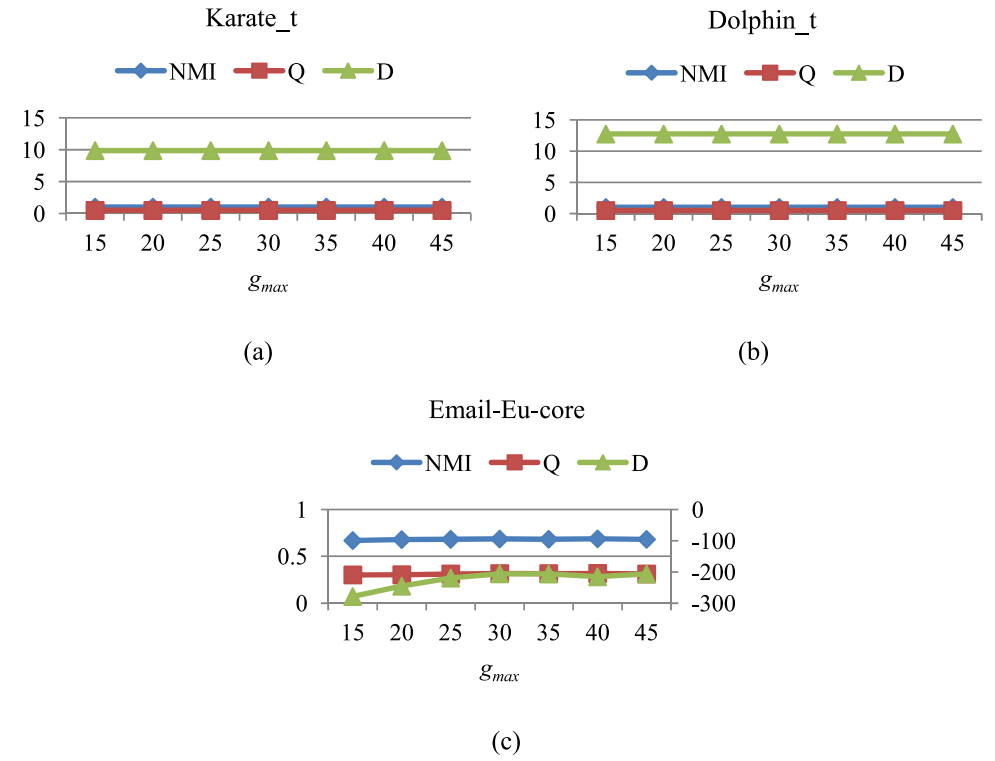


Fig. 8. The effect of parameter g_{max} on three networks. (a) Karate_t. (b) Dolphin_t. (c) Email_Eu_core.

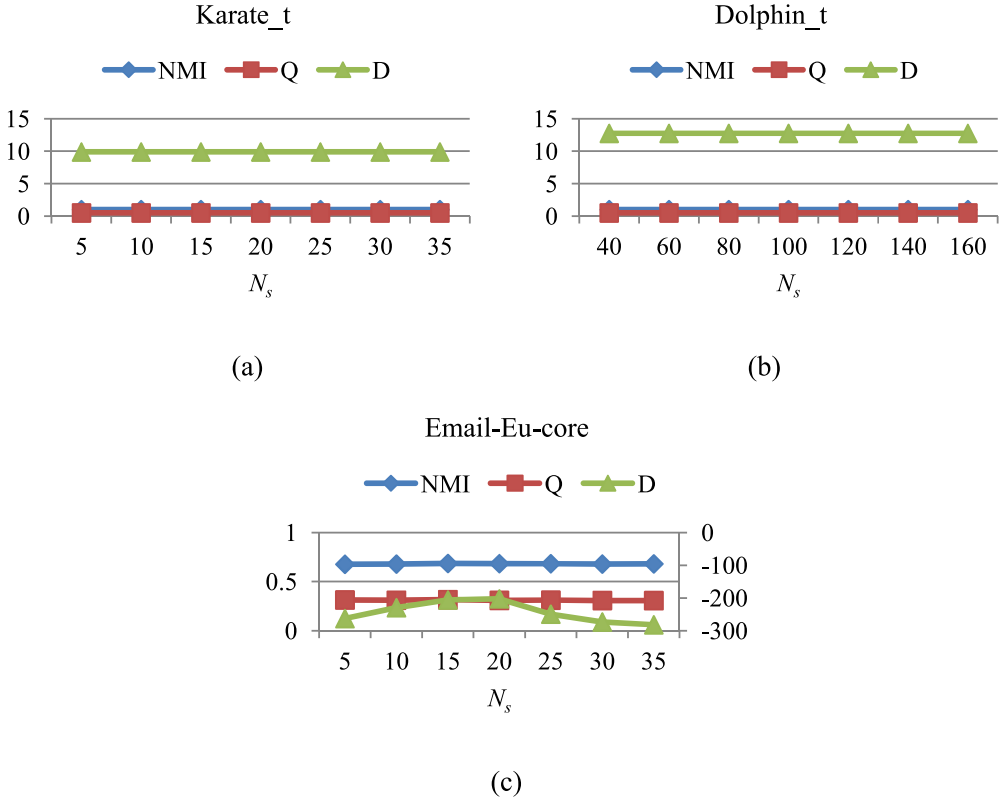


Fig. 9. The effect of parameterN_s on three networks. (a) Karate_t. (b) Dolphin_t. (c) Email_Eu_core.

Email-Eu-core network, respectively. LINE algorithm also obtains the max *NMI* value in Dolphin network. The proposed algorithm can obtain the best detection results on seven networks. Moreover, in Karate_t and Dolphin_t networks containing independent sub-networks, the proposed algorithm can distinguish the independent sub-networks and each community structure very accurately. In the Email-Eu-core network, although the algorithm

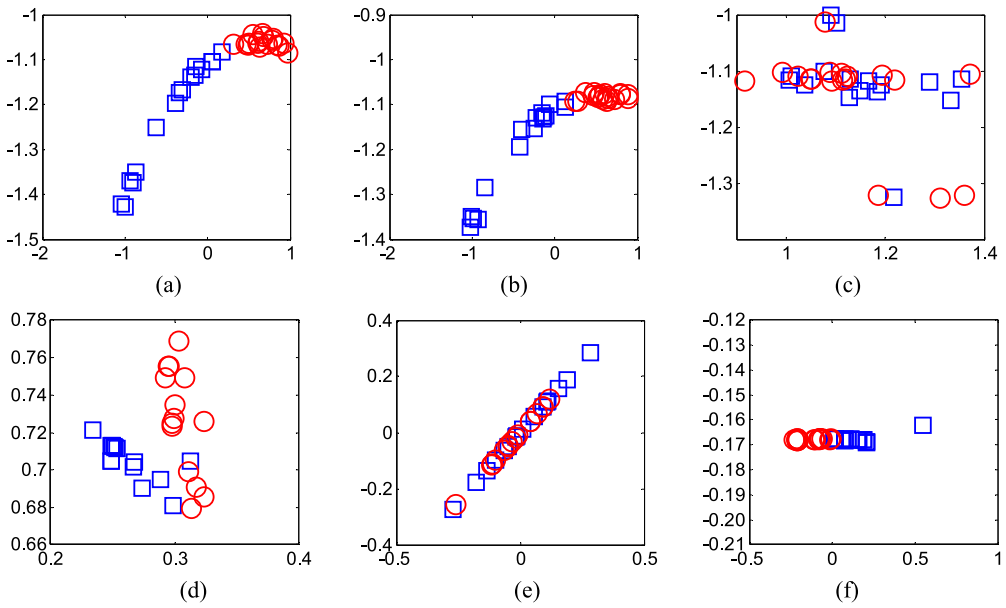


Fig. 10. Visualization results of node embedding on Karate network. (a) Deepwalk. (b) Node2vec. (c) LINE. (d) DNGR. (e) DNE-SBP. (f) SP-MOEA/D.

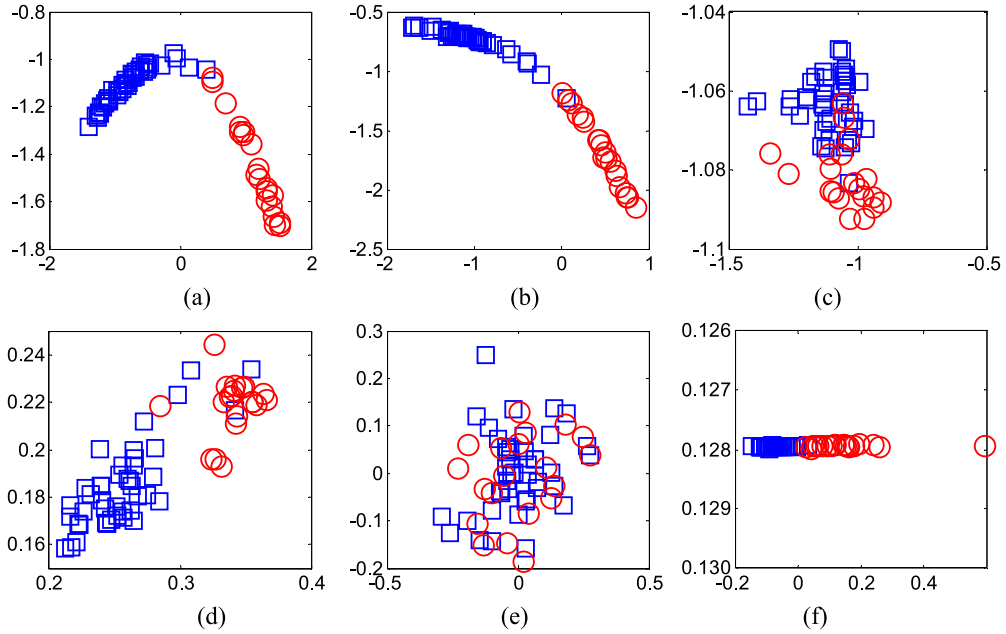


Fig. 11. Visualization results of node embedding on Dolphin network. (a) Deepwalk. (b) Node2vec. (c) LINE. (d) DNGR. (e) DNE-SBP. (f) SP-MOEA/D.

Table 3
NMI values obtained by six algorithms on 8 networks.

NMI	Deepwalk	Node2vec	LINE	DNGR	DNE-SBP	ECD	SP-MOEA/D
Karate	0.4006	0.6766	0.5739	0.6458	0.6499	0.6994	**1**
Dolphin	**0.8888**	0.8141	**0.8888**	0.7531	0.3562	0.5852	**0.8888**
Football	0.9241	0.9268	0.9128	0.7787	0.8706	0.9268	**0.9314**
Polbooks	0.5894	0.5632	0.5629	0.5336	0.5858	0.5639	**0.6557**
Email-Eu-core	**0.7283**	0.7201	0.6874	0.3692	0.2037	0.5620	0.6854
Polblogs	0.4124	0.4248	0.4058	0.0155	0.5092	0.3914	**0.5204**
Karate_t	0.6095	0.5497	0.5241	0.8396	0.7000	0.8570	**1**
Dolphin_t	0.9343	0.9343	0.9343	0.8091	0.7934	0.8038	**1**

does not get the max *NMI* value, compared with DNGR algorithm and DNE-SBP algorithm, the algorithm is still significantly improved.

As shown in Table 4, the network embedding results obtained by six embedding methods are applied to 11 networks. Node2vec algorithm achieves the highest modularity value in Football network and Power network, respectively. Where "-" means that the algorithm cannot get effective community detection results within 10 h. DNE-SBP algorithm achieves the highest modularity value in Polblogs network. Due to its high time complexity, ECD method cannot obtain effective community detection results in

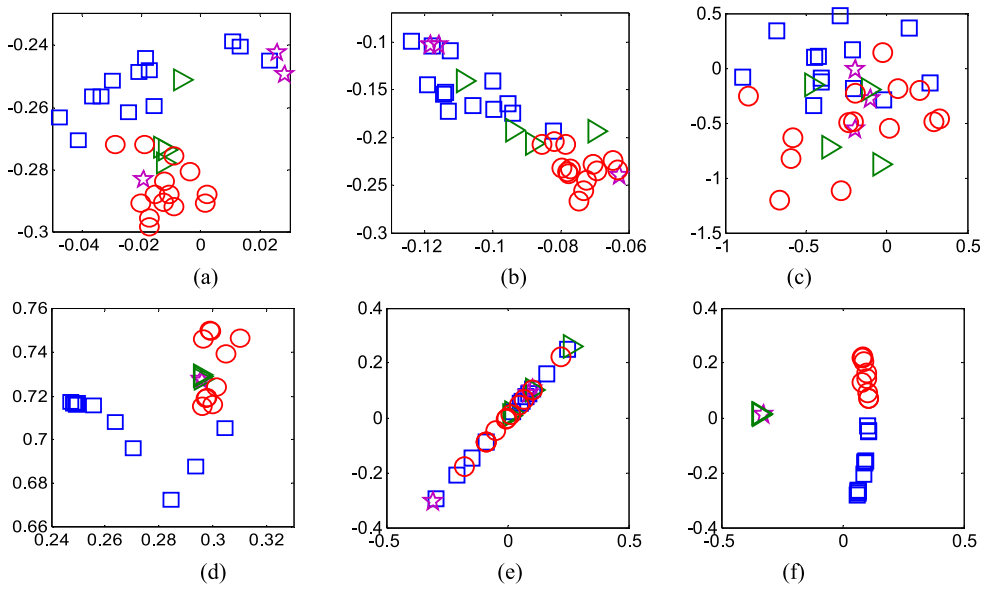


Fig. 12. Visualization results of node embedding on Karate_t network. (a) Deepwalk. (b) Node2vec. (c) LINE. (d) DNGR. (e) DNE-SBP. (f) SP-MOEA/D.

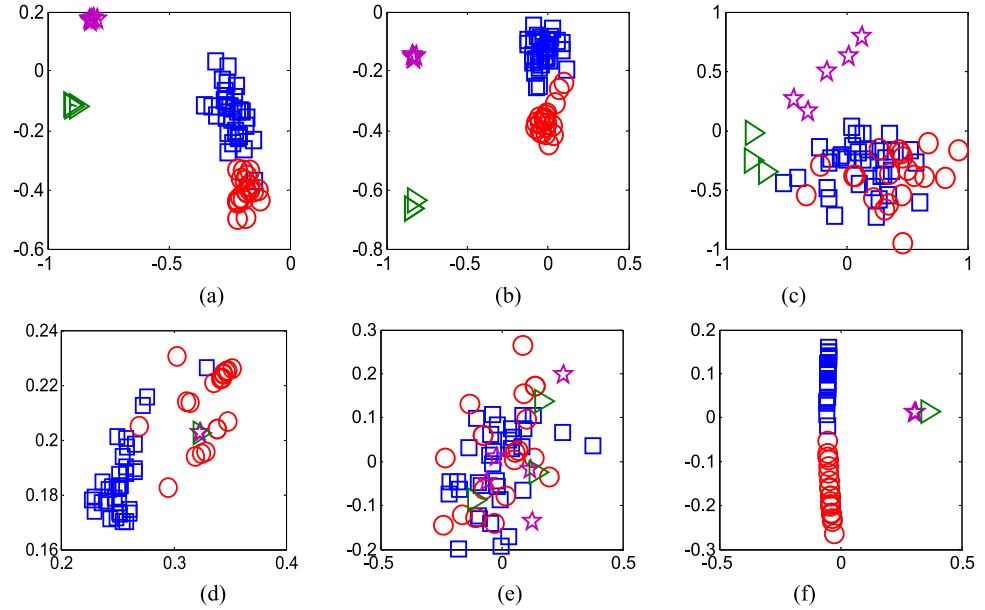


Fig. 13. Visualization results of node embedding on Dolphin_t network. (a) Deepwalk. (b) Node2vec. (c) LINE. (d) DNGR. (e) DNE-SBP. (f) SP-MOEA/D.

a certain period of time when the network scale increases. ECD method obtains the highest Q value in 5 networks.

But reviewing Table 3, the *NMI* value corresponding to ECD method results is not ideal. This is because the modularity function Q, as the evaluation index of community detection results, which can reflect the internal compactness of community structure, but it cannot reflect the most accurate real community structure of the network. The proposed algorithm can get the maximum Q value of modularity on 3 networks. At the same time, it can be seen that although the proposed algorithm does not get the highest modularity value on football network, polblogs network and power network, it is not much different from the highest modularity value. This shows that the algorithm can still get the detection result with closer internal connections of community structure on the basis of getting closer to the real network partition result.

It can be seen from Table 5 that the network embedding results obtained by six embedding methods are applied to 11 networks. Where "-" means that the algorithm cannot get effective community detection results within 10 h. The node2vec algorithm achieves the highest modularity density value on power network. ECD method obtains the highest$D$ value on six networks. In the same way as described in the previous paragraph, the modularity density can reflect the internal compactness of the community structure, and improve the resolution limitation of the modularity function when the network scale increases, but it still has limitations in reflecting the most accurate real community structure of the network. The proposed algorithm can get the maximum modularity density on 3 networks. This shows that the proposed algorithm can still get the detection result with closer internal connections of community structure on the basis of getting closer to the real network partition result.

Wilcoxon signed rank test is used to analyze the results of the proposed algorithm and 6 comparative algorithms. The Wilcoxon signed rank test results of the proposed algorithm and the comparison algorithms are shown in Tables 6–8. Where SF represents the proposed algorithm.

Table 4
Q value of community detection on 11 networks.

Q	Deepwalk	Node2vec	LINE	DNGR	DNE-SBP	ECD	SP-MOEA/D
Karate	0.2524	0.3205	0.3123	0.3132	0.3132	**0.402**	0.3715
Dolphin	0.3787	0.3688	0.3698	0.3898	0.2772	**0.5202**	0.3926
Football	0.6005	**0.6010**	0.5935	0.4648	0.5384	0.5932	0.5958
Polbooks	0.5012	0.5151	0.5189	0.4641	0.5118	0.5122	**0.5204**
Email-Eu-core	0.2929	0.2925	0.2771	0.0591	0.0292	0.095	**0.3166**
Polblogs	0.1423	0.1342	0.1357	0.0001	**0.4230**	0.4159	0.4180
Karate_t	0.3619	0.3771	0.3771	0.4182	0.2615	0.4817	**0.4880**
Dolphin_t	0.5429	0.5293	0.5376	0.441	0.4192	**0.5662**	0.4868
SFI	0.7191	0.7213	0.7114	0.7189	0.2439	**0.7476**	0.7263
Netscience	0.2121	0.2101	0.2099	0.6568	0.5271	**0.9494**	0.6670
Power	0.9259	**0.9290**	0.9255	0.4162	0.5184	–	0.9075

Table 5
D value obtained by community detection on 11 networks.

D	Deepwalk	Node2vec	LINE	DNGR	DNE-SBP	ECD	SP-MOEA/D
Karate	4.7059	5.9028	5.8246	6.1391	6.1391	**7.845**	6.833
Dolphin	9.0750	8.5068	9.0750	8.9030	6.2643	**11.461**	9.0750
Football	41.846	41.210	40.482	4.4034	25.713	40.185	**41.997**
Polbooks	17.686	18.005	18.329	14.569	15.542	18.487	**18.595**
Email-Eu-core	−432.79	−413.06	−542.78	−1024.5	−386.16	−8850.1	**−205.71**
Polblogs	8.9594	8.2847	8.3394	27.231	39.430	**49.677**	37.753
Karate_t	4.0278	4.3024	4.5960	7.9167	1.3376	9.8002	**9.8828**
Dolphin_t	12.762	12.762	10.762	7.8258	8.3124	**14.472**	12.747
SFI	16.070	16.225	15.741	15.898	−2.3427	**25.304**	16.362
Netscience	−785.94	−823.37	−807.69	78.83	57.287	**714.48**	87.978
Power	96.949	**97.074**	96.256	−226.95	21.493	–	50.455

Table 6
Wilcoxon signed rank test results of *NMI* values of the proposed algorithm and comparison algorithms.

NMI	SF/Deepwalk		SF/Node2vec		SF/LINE		SF/DNGR		SF/DNE-SBP		SF/ECD	
	p	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*
Karate	0.000046	1	0.000016	1	0.000055	1	0.00004	1	0.000033	1	0.000046	1
dolphin	0.014801	1	1.000000	0	0.724189	0	1.000000	0	0.000123	1	0.01615	1
football	0.000063	1	0.000063	1	0.000063	1	0.000064	1	0.000064	1	0.000062	1
polbooks	0.005102	1	0.000155	1	0.000163	1	0.00006	1	0.002119	1	0.00006	1
email-Eu-core	0.000183	1	0.000183	1	0.909688	0	0.000183	1	0.000183	1	0.000179	1
polblogs	0.000162	1	0.000064	1	0.000064	1	0.000064	1	0.729792	0	0.000178	1
Karate_t	0.000033	1	0.000024	1	0.00006	1	0.000057	1	0.000063	1	0.000016	1
dolphin_t	0.000061	1	0.000049	1	0.000063	1	0.000055	1	0.000064	1	0.000046	1

Table 7
Wilcoxon signed rank test results of *Q* values of the proposed algorithm and comparison algorithms.

Q	SF/Deepwalk		SF/Node2vec		SF/LINE		SF/DNGR		SF/DNE-SBP		SF/ECD	
	p	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*
Karate	0.000046	1	0.000016	1	0.000055	1	0.00004	1	0.000033	1	0.000046	1
dolphin	1.000000	0	1.000000	0	1.000000	0	1.000000	0	0.00011	1	0.000047	1
football	0.001421	1	0.017092	1	0.000063	1	0.000064	1	0.000064	1	0.001401	1
polbooks	0.728679	0	0.788122	0	0.098237	0	0.000055	1	0.101112	0	0.016949	1
email-Eu-core	0.000183	1	0.000183	1	0.000183	1	0.000182	1	0.000183	1	0.000179	1
polblogs	0.000076	1	0.000024	1	0.000024	1	0.000024	1	0.000066	1	0.000137	1
Karate_t	0.000033	1	0.000024	1	0.00006	1	0.000057	1	0.000063	1	0.000016	1
dolphin_t	0.016902	1	0.015505	1	0.114946	0	0.000055	1	0.000064	1	0.000046	1
SFI/6	1.000000	0	1.000000	0	1.000000	0	1.000000	0	0.000145	1	0.002383	1
netscience/396	0.000181	1	0.00018	1	0.000181	1	0.307125	0	0.000181	1	0.000176	1
power/40	0.000181	1	0.000181	1	0.000181	1	0.000181	1	0.000181	1	–	–

The *p* value represents the probability that the median value of two samples is equal. When *p* is close to 0, the null hypothesis should be questioned. The *h* value represents the test result. When $h = 0$, it means that the median difference between the two samples is not significant. While $h = 1$ means that the difference between the median of two samples is significant. The ECD algorithm cannot get effective community detection results within 10 h. Compared Tables 6–8 with Tables 3–5, the proposed algorithm can obtain the highest *NMI* value on 7 networks, and the Wilcoxon signed rank test results show that most of the differences between the proposed algorithm and the comparison algorithms are obvious. Although there are not many networks with the highest *Q* and*D* values obtained by the proposed algorithm, the differences of the improvements are obvious compared with all the algorithms. This further verifies the effectiveness of the proposed algorithm.

5. Conclusion

A complex network graph embedding method (SP-MOEA/D) based on shortest path matrix and decomposition multi-objective evolutionary algorithm is proposed, which can better reflect the network structure at the level of community structure in network. This paper presents a network embedding method based on node distance matrix. By calculating the shortest path matrix

Table 8
Wilcoxon signed rank test results of D values of the proposed algorithm and comparison algorithms.

D	SF/Deepwalk		SF/Node2vec		SF/LINE		SF/DNGR		SF/DNE-SBP		SF/ECD	
	p	h	p	h	p	h	p	h	p	h	p	h
Karate	0.000046	1	0.000016	1	0.000055	1	0.00004	1	0.000033	1	0.000046	1
dolphin	0.014607	1	1.000000	0	0.723867	0	1.000000	0	0.000121	1	0.000053	1
football	0.000064	1	0.000063	1	0.000063	1	0.000064	1	0.000064	1	0.000062	1
polbooks	0.02334	1	0.012818	1	0.378325	0	0.000058	1	0.000168	1	0.44014	0
email-Eu-core	0.000183	1	0.000183	1	0.000183	1	0.000183	1	0.000183	1	0.000178	1
polblogs	0.000161	1	0.000063	1	0.000063	1	0.000063	1	0.000144	1	0.000177	1
Karate_t	0.000033	1	0.000024	1	0.00006	1	0.000057	1	0.000063	1	0.000016	1
dolphin_t	0.114321	0	0.43572	0	0.001407	1	0.000055	1	0.000064	1	0.000046	1
SFI/6	0.468465	0	0.731586	0	1.000000	0	0.340125	0	0.000161	1	0.000054	1
netscience/396	0.000182	1	0.000182	1	0.000182	1	0.427181	0	0.140316	0	0.000173	1
power/40	0.000181	1	0.000182	1	0.000245	1	0.000181	1	0.000178	1	–	–

and the similarity between nodes in the network, the network node relationship matrix reflecting the network topology and the degree of node tightness is obtained. The network similarity matrix is further calculated, and then the low-dimensional vector representation of nodes is obtained by random surfing and multi-dimensional scaling. Then, the community structure of the network can be detected based on the obtained node representation structure. Starting from the essence of network structure and the tightness between nodes, this method can reflect the relationship characteristics of network nodes more effectively, and then obtain the vector representation of network nodes which can more accurately reflect the information of community structure. Two objective functions are designed, which are combined with the network core nodes (potential community structure center). So that the low-dimensional vector representation results of the nodes can better reflect the network structure and the tightness between nodes at the community structure level. Furthermore, a decomposition based multi-objective optimization method is proposed to allocate distances to unrelated nodes for community detection in disconnected networks. This method can better reflect the community structure information in the disconnected networks. The proposed algorithm still needs to be improved. For example, the complexity of evolutionary optimization algorithm is high, so it has less advantages in dealing with large-scale network data sets. In the future research work, we will focus on the larger scale of complex network embedding problem to solve the more complex community detection problem.

CRediT authorship contribution statement

Weitong Zhang: Design (methodology), Investigation (performing experiments or data/evidence collection), Drafting and revising the article. **Ronghua Shang:** Revising the article, Supervision, Project administration. **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.

Acknowledgments

We would like to express our sincere appreciation to the editors and the anonymous reviewers for their insightful comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Natural Science Foundation of China under Grants 61773304, 61871306, 61772399, 61836009, and U1701267, the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) under Grants No. B07048, the Major Research Plan of the National Natural Science Foundation of China under Grants 91438201 and 91438103, and the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT1170.

References

[1] R. Angles, C. Gutierrez, Survey of graph database models, ACM Comput. Surv. 40 (1) (2008) 1.
[2] W. Zhang, R. Zhang, R. Shang, J. Li, L. Jiao, Application of natural computation inspired method in community detection, Physica A (2018).
[3] J. Gao, J.L. Gao, A similarity measurement method based on graph kernel for disconnected graphs, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI, 2019.
[4] W.L. Hamilton, R. Ying, J. Leskovec, Representation learning on graphs: Methods and applications, 2017, arXiv preprint arXiv:1709.05584.
[5] L. Ma, M. Gong, J. Liu, Q. Cai, L. Jiao, Multi-level learning based memetic algorithm for community detection, Appl. Soft Comput. 19 (2014) 121–133.
[6] A. Said, R.A. Abbasi, O. Maqbool, A. Daud, CC-GA: A clustering coefficient based genetic algorithm for detecting communities in social networks, Appl. Soft Comput. 63 (2018) 59–70.
[7] S. Bhagat, G. Cormode, S. Muthukrishnan, Node classification in social networks, in: Social Network Data Analytics, Springer, Boston, MA, 2011, pp. 115–148.
[8] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Phys. A 390 (6) (2011) 1150–1170.
[9] R. Shang, H. Liu, L. Jiao, A.M. Ghalamzan E., Community mining using three closely joint techniques based on community mutual membership and refinement strategy, Appl. Soft Comput. 61 (2017) 1060–1073.
[10] R. Shang, W. Zhang, L. Jiao, R. Stolkin, Y. Xue, A community integration strategy based on an improved modularity density increment for large-scale networks, Physica A 469 (2017) 471–485.
[11] M. Gong, L. Ma, Q. Zhang, L. Jiao, Community detection in networks by using multiobjective evolutionary algorithm with decomposition, Physica A 391 (15) (2012) 4050–4060.
[12] R. Shang, S. Luo, W. Zhang, R. Stolkin, L. Jiao, A multiobjective evolutionary algorithm to find community structures based on affinity propagation, Physica A 453 (2016) 203–227.
[13] Y. Guo, H. Yang, M. Chen, et al., Ensemble prediction-based dynamic robust multi-objective optimization methods, Swarm Evol. Comput. 48 (2019) 156–171.
[14] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (23) (2006) 8577–8582.
[15] V. Zlatić, A. Gabrielli, G. Caldarelli, Topologically biased random walk and community finding in networks, Phys. Rev. E 82 (2010) 066109.
[16] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[17] M. Belkin, P. Niyogi, Laplacian Eigenmaps and spectral techniques for embedding and clustering, Adv. Neural Inf. Process. Syst. (2002) 585–591.
[18] D. Luo, F. Nie, H. Huang, C.H. Ding, Cauchy graph embedding, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 553-560.
[19] M. Ou, P. Cui, J. Pei, Z. Zhang, W. Zhu, Asymmetric transitivity preserving graph embedding, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1105–1114.
[20] L. Lovász, Random walks on graphs: A survey, Combinatorics 2 (1) (1993) 1–46, Paul erdos is eighty.

[21] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 701–710.
[22] C. McCormick, Word2vec tutorial-the skip-gram model, 2016.
[23] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864.
[24] H. Chen, B. Perozzi, Y. Hu, S. Skiena, Harp: Hierarchical representation learning for networks, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[25] W. Gu, L. Gong, X. Lou, J. Zhang, The hidden flow structure and metric space of network embedding algorithms based on random walks, Sci. Rep. 7 (1) (2017).
[26] D. Wang, P. Cui, W. Zhu, Structural deep network embedding, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1225–1234.
[27] V. Derhami, E. Khodadadian, M. Ghasemzadeh, Applying reinforcement learning for web pages ranking algorithms, Appl. Soft Comput. 13 (4) (2013) 1686–1692.
[28] S. Cao, W. Lu, Q. Xu, Deep neural networks for learning graph representations, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[29] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Line: Large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.
[31] S. Hougardy, The Floyd–Warshall algorithm on graphs with negative cycles, Inform. Process. Lett. 110 (8–9) (2010) 279–281.
[32] I. Borg, P. Groenen, Modern multidimensional scaling: Theory and applications, J. Educ. Meas. 40 (3) (2003) 277–280.
[33] J. Ye, Cosine similarity measures for intuitionistic fuzzy sets and their applications, Math. Comput. Modelling 53 (1–2) (2011) 91–97.
[34] R. Rossi, N. Ahmed, The network data repository with interactive graph analytics and visualization, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[35] R.H. Shang, W.T. Zhang, L.C. Jiao, X.R. Zhang, R. Stolkin, Dynamic immunization node model for complex networks based on community structure and threshold, IEEE Trans. Cybern. (2020) Accepted.
[36] Z. Liao, W. Gong, X. Yan, L. Wang, C. Hu, Solving nonlinear equations system with dynamic repulsion-based evolutionary agorithms, IEEE Trans. Syst. Man Cybern. 50 (4) (2020) 1590–1601.
[37] W. Gong, Z. Cai, Parameter extraction of solar cell models using repaired adaptive differential evolution, Sol. Energy 94 (2013) 209–220.
[38] F. Hajabdollahi, Z. Hajabdollahi, H. Hajabdollahi, Soft computing based multi-objective optimization of steam cycle power plant using NSGA-II and ANN, Appl. Soft Comput. 12 (11) (2012) 3648–3655.
[39] I. Aydin, M. Karakose, E. Akin, A multi-objective artificial immune algorithm for parameter optimization in support vector machine, Appl. Soft Comput. 11 (1) (2011) 120–129.
[40] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[41] Y. Guo, X. Zhang, D. Gong, et al., Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks, IEEE Trans. Evol. Comput. 24 (04) (2020) 750–764.
[42] X. Shen, F.L. Chung, Deep network embedding for graph representation learning in signed networks, IEEE Trans. Cybern. (2018).
[43] F.Z. Liu, J. Wu, C. Zhou, J. Yang, Evolutionary community detection in dynamic social networks, in: International Joint Conference on Neural Networks, Budapest, Hungary, IJCNN 2019.
[44] H. Yin, A.R. Benson, J. Leskovec, D.F. Gleich, Local higher-order graph clustering, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 555–564.
[45] R. Rossi, N. Ahmed, The network data repository with interactive graph analytics and visualization, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[46] Y. Sun, M. Kirley, S.K. Halgamuge, Quantifying variable interactions in continuous optimization problems, IEEE Trans. Evol. Comput. 21 (2) (2016) 249–264.
[47] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[48] Z. Li, S. Zhang, R.S. Wang, X.S. Zhang, L. Chen, Quantitative function for community detection, Phys. Rev. E 77 (2008) 036109.