Density-Peak-Based Overlapping Community Detection Algorithm

Liping Sun, Tao Ye, Jian Sun, Xiaoyu Duan, and Yonglong Luo

***Abstract*— Overlapping community detection is essential for revealing the hidden structure of complex networks. In this work, we present an overlapping community detection algorithm that selects community centers adaptively based on density peaks. The proposed algorithm, called the density-peak-based overlapping community detection (DPOCD) algorithm, defines point link strength and edge link strength to construct distance matrix. Unlike the density peaks clustering algorithm, by which cluster centers are selected manually, the DPOCD algorithm uses the linear fitting method to select community centers. To evaluate the feasibility of the presented algorithm, we compared it with other advanced methods on artificial synthetic network and real complex network datasets. The experimental results demonstrate that our method achieves excellent performance in large-scale complex networks and the robustness of the algorithm.**

***Index Terms*— Adaptive selection of cluster centers, community detection algorithm, density peak, overlapping community.**

I. Introduction

With the popularity of the Internet of Things and the rapid development of social information networks, the detection of overlapping community structures in real-world complex networks is meaningful for measuring the structures and properties of networks [1]. Community structure is a fundamental characteristic of complex networks. The nodes in the same community have close link relationships, whereas nodes in different communities have sparse link relationships [2]. The important objective of community detection is to reveal the hidden structure of real-world complex network [3]. The traditional community detection algorithms assume that a node can belong to only one community, but it is possible for some nodes in a complex network abstracted from a real system to belong to more than one community [4]. Overlapping communities are those that have overlapping nodes in their structure. The larger the number of overlapping nodes, the higher the degree of overlap among the communities [5].

Rodriguez and Laio [6] proposed the density peaks clustering (DPC) algorithm, which is designed based on the idea that the cluster centers have higher density than the surrounding nodes and are farther away from other cluster centers. It overcomes the disadvantages of the density-based spatial clustering of applications with noise (DBSCAN) algorithm [7], which requires specification of the threshold and the neighborhood radius. The community detection methods based on DPC have further advantages of efficient node allocation and the ability to detect nonspherical communities. However, whereas communities in social networks are often overlapped, DPC only detects nonoverlapping communities. Moreover, it needs to select cluster centers manually, which works well for smaller datasets but is unrealistic when the dataset is large.

Based on the definitions of point link strength and edge link strength, this article proposes the density-peak-based overlapping community detection (DPOCD) algorithm. Using improved DPC method, the DPOCD algorithm is implemented to detect overlapping communities in complex networks. The main work and innovations are as follows.

1) DPOCD algorithm considers the influence of link relationships, node degrees, and common nodes on link strength, defines point link strength and edge link strength, and calculates the distance matrix according to the definition of link strength.
2) DPOCD algorithm incorporates linear fitting to select the community centers adaptively. It makes the results of community center selection more objective and thereby overcomes the strong subjectivity of the results of DPC due to its strategy of manually selecting the cluster centers.
3) In contrast to the assignment mechanism of DPC, the DPOCD algorithm uses a probability vector to represent the degree of belonging of nodes to each community, which is determined by the k nodes whose local densities are higher than that of the current node and whose distance from the current node is shortest.

The remainder of this article is organized as follows. In Section II, related work is discussed. In Section III, the main idea of the DPC algorithm is reviewed. The details of the proposed DPOCD algorithm are presented in Section IV. Section V introduces the experimental evaluation criteria and analyzes the results of the experiments conducted on synthetic

Manuscript received 23 May 2021; revised 10 September 2021; accepted 8 October 2021. Date of publication 1 November 2021; date of current version 2 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61972439 and Grant 61602009, in part by the Key Program in the Youth Elite Support Plan in Universities of Anhui Province under Grant gxyqZD2020004, in part by the Anhui Provincial Natural Science Foundation of China under Grant 2108085MF214, and in part by The University Synergy Innovation Program of Anhui Province under Grant GXXT-2021-007. *(Corresponding author: Yonglong Luo.)*

Liping Sun, Tao Ye, Xiaoyu Duan, and Yonglong Luo are with the School of Computer and Information, Anhui Provincial Key Laboratory of Network and Information Security, Anhui Normal University, Wuhu 241003, China (e-mail: slp620@163.com; 1242338985@qq.com; 992493123@qq.com; ylluo@ustc.edu.cn).

Jian Sun is with the Software Institute, Nanjing University, Nanjing 210023, China (e-mail: jiansun@smail.nju.edu.cn).

Digital Object Identifier 10.1109/TCSS.2021.3122018

and real datasets. Finally, conclusions and future work are given in Section VI.

II. Related Work

A. Traditional Overlapping Community Detection Algorithm

The traditional overlapping community detection algorithms can be divided into three categories, namely, algorithms based on faction filtering and multiobjective evolutionary algorithm (MOEA), algorithms based on label propagation, and algorithms based on local expansion and optimization.

1) Algorithms Based on Faction Filtering and MOEA: Based on the clique percolation method (CPM) proposed by Palla *et al.* [5] in 2005, the fully connected graph composed of k nodes was defined as k factions. If the factions shared $k-1$ nodes, then they were considered to be connected and could be merged. Simultaneously, the shared nodes among the disconnected factions were regarded as the overlapping nodes among the communities which the factions belonged to. Based on the mechanism of fully connected graph, a small number of nodes in the network not belonging to any community was achieved. Maity and Rath [8] proposed the extended clique percolation method (ECPM) to solve this problem. First, the initial community in the network was found according to the CPM algorithm, and then the unprocessed nodes were assigned to each community by calculating the attribution coefficient. Shen *et al.* [9] proposed an algorithm combined with clique filtering, which took maximal clique as the basic unit of the community, and detected community in the network based on the aggregation method. The algorithm could effectively detect overlapping and layered communities. At the same time, the algorithm introduced the quality of the community detected by the modular function *EQ* degree measurement method. Nicosia *et al.* [10] also extended the modularity function and proposed a new modularity function Q_{ov}, which could effectively measure the quality of overlapping communities in directed networks. To improve the detection quality of the CPM algorithm, Qian *et al.* [11] improved the CPM algorithm based on maximum clique similarity, and the improved algorithm could detect overlapping communities in large-scale complex networks. Esmaeili *et al.* [12] proposed an effective semidefinite programming (SDP) community detection solution, which combined nongraph data, which were hereinafter referred to as edge information. SDP was an effective solution for standard community detection of graphs. The simulation results showed that the asymptotic results in this article could also provide a basis for the SDP performance of medium-sized graphs. Wen *et al.* [13] proposed a maximal clique-based MOEA for overlapping community detection. In this algorithm, a new representation scheme based on the introduced maximal clique graph was presented; the new representation scheme allowed MOEAs to handle the overlapping community detection problem in a way similar to that of the separated community detection, such that the optimization problems were simplified. Zhang *et al.* [14]. proposed a mixed-representation-based MOEA (MR-MOEA) for overlapping community detection. In MR-MOEA, a mixed individual representation scheme was proposed to fast encode and decode the overlapping divisions of complex networks. The effectiveness of the proposed algorithm was verified on a real dataset.

2) Algorithms Based on Label Propagation: The traditional label propagation algorithm [15] assigned a unique label to each node of network. Then in the iterative process, it took the label with the largest number of occurrences in the neighborhood as the new label of the node. The traditional label propagation algorithm had the advantages of linear time complexity and simple logic, but it was only suitable for detecting nonoverlapping communities in complex networks. Community overlap propagation algorithm (COPRA) [16] was an extension of the label propagation algorithm for detecting overlapping communities of complex networks. First, labels were assigned to each node in the network by group, and then the labels were propagated iteratively according to the similarity until the size of the remaining labels in the network kept unchanged at two consecutive iterations. It had the advantages of high efficiency, but the operation result was unstable and poor robustness. Speaker–listener label propagation algorithm (SLPA) proposed by Xie and Szymanski [17] updated the label according to the speaker–listener policy and defined the label sequence to store the tags updated each time. The experimental results showed that SLPA could accurately identify the overlap structure of node level and community level. Alghamdi and Greene [18] proposed the pairwise constrained SLPA (PC-SLPA) algorithm, which encoded external information into pairwise constraints by semisupervision strategy and detected the community by label propagation, which overcame the disadvantage that the unsupervised label propagation algorithm could not recognize the community structure of specific significance. Lu *et al.* [19] proposed the label propagation algorithm with neighbor node influence (LPANNI) method, which calculated the importance of nodes to determine the label propagation sequence and propagated labels according to the neighbor influence strategy and historical label priority strategy, which improved the accuracy and stability of label based on the propagation algorithm in large-scale complex networks. Huang *et al.* [20] proposed the node-similarity-based multilabel propagation algorithm (NMLPA), which first calculated the similarity between nodes according to attributes, and then propagated multiple labels according to node similarity and topological sequence. The NMLPA algorithm overcame the disadvantage of random selection of labels based on the label propagation algorithm and applied pruning strategy to control the number of node labels to enhance the stability of the algorithm. Gao *et al.* [21] proposed an overlapping community detection algorithm based on membership propagation, which used the global and local information of the nodes. The concept of membership was introduced, which not only stored label information but also stored the membership of the belonging nodes. Label-based methods were efficient in community detection, but the community structure they discover was highly uncertain [22].

3) Algorithms Based on Local Expansion and Optimization: The traditional community detection algorithms were mostly global oriented, that is, the algorithm needed to predict the global topology of complex networks. However, with the

rapid development of society, the scale of complex networks abstracted by the real system expands rapidly, and the community detection algorithm based on global strategy could not effectively identify the communities in complex networks. Lancichinetti *et al.* [23] proposed a method based on the local optimization of a fitness function (LFM), which adopted the strategy of local expansion and optimization. In the beginning, the seed node was randomly selected, and then the node was iteratively selected based on the neighborhood to join the community whose seed node was located until the fitness function no longer changes. The algorithm had high efficiency, but it had the disadvantages of strong randomness and needed to customize the fitness function parameters. Whang *et al.* [24] proposed the neighborhood-inflated seed expansion (NISE) algorithm. It selected seed nodes by graph clustering center and propagation center strategy, and then expanded the community of seed nodes according to *PageRank* technology. The results of the experiment showed that it could find overlapping clusters in the graph. Jian *et al.* [25] proposed community detection by the local structure expansion (CLOSE) algorithm, which initially selected a set of seed nodes based on the link function, and then propagated the locally expanded seed nodes based on the label. Liu *et al.* [26] proposed an overlapping community discovery algorithm based on the idea of local optimal extended cohesion, constructed the initial core community with the most important nodes and their neighbors, and expanded the core community by the degree of node attributes to guide the algorithm to meet the termination conditions. Yu *et al.* [27] proposed an overlapping community detection algorithm based on random walk and seed expansion, which used a random walk strategy to find seed communities with a compact structure. The disadvantage of these methods was that the clustering results were unstable because the seeds were randomly selected. Berahmand *et al.* [28] proposed a local method based on core node detection and extension. First, based on the similarity between nodes in the graph, community center nodes (core nodes) with high embeddability were detected. Then, the expansion of these nodes would be considered, using the concept of node's membership based on the definition of strong community for weighted graphs. Cheng *et al.* [29] proposed an efficient local-expansion-based overlapping community detection algorithm used local neighborhood information (OCLN). During the iterative expansion process, only neighbors of nodes added in the last iteration (rather than all neighbors) were considered to determine whether they could join the community. A belonging coefficient was also proposed in OCLN to filter out incorrectly identified nodes.

B. Cluster-Based Overlapping Community Detection Algorithm

Rodriguez and Laio [6] proposed the density peak clustering algorithm, which had the advantages of high node allocation efficiency, without the need to specify the number of clusters and could detect nonspherical clusters. Scholars tried to apply the idea of the DPC algorithm to detect community in complex networks. Bai *et al.* [30] proposed an overlapping community detection algorithm based on density peaks (OCDDP), which took the number of paths that could be reached by specified hops between nodes as the distance between nodes and used probability vector to represent the possibility that nodes belonged to different communities. The algorithm had strong robustness, but it did not perform well in detecting community structure in large-scale networks. Xu *et al.* [31] proposed an overlapping community detection algorithm, which considered the influence of common nodes on the distance between nodes and used the concept of wrapping node to design the remaining node allocation algorithm. The algorithm had the advantage of detecting community structure in weighted and unweighted networks and had better performance in networks with complex weight distribution. Lu *et al.* [32] proposed a community detection algorithm, which used the improved DPC method to obtain the number of centers as a preallocation parameter for nonnegative matrix decomposition, to solve the problem that most nonnegative matrix factorization-based community detection algorithms needed to determine the number of communities in advance. Wang and Xu [33] proposed a clustering method based on adaptive density peak detection, estimated local density by nonparametric multivariate kernel estimation, and selected cluster centers by an automatic clustering method based on maximizing the average contour index. Jiang *et al.* [34] proposed the density fragment clustering algorithm, inspired by DPC and DBSCAN, which improved its ability on various complex datasets by combining density fragments based on network structure similarity.

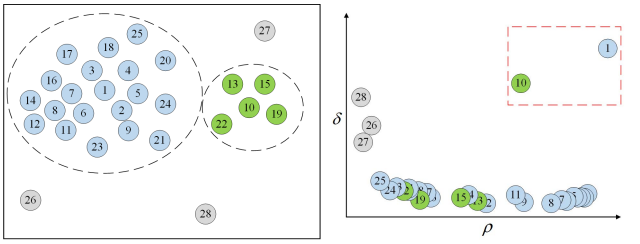
Detection of overlapping communities in complex networks is one of the hot topics in machine learning. Some algorithms do not perform well in dealing with large-scale complex networks, while others have the disadvantage of high time complexity. The DPOCD algorithm proposed in this article defines the concepts of point link strength and edge link strength, calculates the distance matrix, and then realizes the automatic selection of cluster centers of overlapping communities by combining with the linear fitting method. Simultaneously, the algorithm defines the probability vector to represent the degree of belonging of nodes to each community and realizes the allocation of the remaining nodes. Experiments indicate that the DPOCD algorithm can accurately detect overlapping communities in large-scale networks. It also shows the robustness of the DPOCD algorithm. It is an overlapping community detection algorithm with low time complexity and high recognition accuracy.

III. Density Peak Clustering Algorithm

The DPOCD algorithm is designed based on the DPC algorithm [6]. Here is a brief introduction to its central concept.

The DPC algorithm considers that the local density of a cluster center is higher than that of the nodes in its neighborhood and that the cluster center is far from nodes whose local density is higher. In the algorithm, ρ_i denotes the local density of node i, and δ_i denotes the distance between node i and the closest node having a higher density. The local density ρ_i and the shortest distance δ_i are given by the following equation:

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right) \tag{1}$$



(a) Complex network data distribution (b) ρ-δ decision diagram

Fig. 1. Schematic of the DPC algorithm: (a) node distribution diagram and (b) ρ–δ decision diagram. As shown in (b), both ρ and δ of nodes 1 and 10 are large. Therefore, nodes 1 and 10 are selected as the cluster centers.

$$\chi(x) = \begin{cases} 1, & \text{if } x < 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} \text{dist}_{ij}. \tag{3}$$

In (1)–(3), dist_{ij} represents the Euclidean distance from node i to node j, and d_c represents the truncation distance. In special cases, when the local density of node t is the maximum globally, its shortest distance δ_t is defined as $\max_j \text{dist}_{ij}$.

After calculating the local density ρ and the shortest distance δ, DPC manually selects the nodes having large ρ and δ values from the ρ–δ decision diagram as the cluster centers. The horizontal coordinate of the decision graph is the local density ρ, and the vertical coordinate is the shortest distance δ, as shown in Fig. 1.

Finally, according to the residual node allocation mechanism of DPC, each of the remaining nodes is allocated to the cluster with the highest local density and the closest clustering center.

When calculating the density on smaller datasets, using (1) and (2) may result in large errors; therefore, an improved Gaussian kernel [6], [35] can be used to calculate the density, as shown in the following equation:

$$\rho_i = \sum_j \exp\left(-\left(\frac{\text{dist}_{ij}}{d_c}\right)^2\right). \tag{4}$$

IV. Proposed Algorithm: DPOCD

To fully use the advantages of DPC and improve the performance of the overlapping community detection algorithm, an overlapping community detection algorithm based on density peaks is proposed. The algorithm is suitable for both weighted and unweighted networks. Sections IV-A–D describe the four main aspects of the algorithm: 1) its representation of the distance between nodes, 2) its methods of calculating ρ and δ for complex networks, 3) its adaptive selection of community centers, and 4) its optimization of the residual node allocation mechanism of the DPC algorithm to find overlapping nodes.

Different from the other algorithms to calculate the distance matrix by link strength, our algorithm defines point link strength and edge link strength to calculate the distance matrix and selects the cluster center by linear fitting. Through the complexity analysis, we verify that the complexity of our algorithm is lower than that of other clustering-based algorithms, and it can also show a better performance in the experimental part.

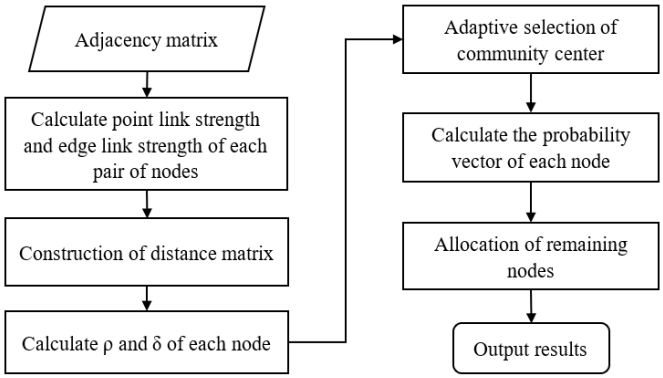


Fig. 2. DPOCD algorithm framework.

TABLE I
Notation Used in DPOCD Algorithm

Notation	Description
A	Adjacency matrix
V_{ij}	Common node set of node i and node j
w_i	The sum of the weights of the edges incident to vertex i
r	Range of the node weights
ρ	Density of nodes
δ	Shortest distance to higher density
ρ^*	Local density after normalization
δ^*	Shortest distance after normalization
γ	Product of ρ^* and δ^*
p_i	Probability vector of node i
w_{ij}	Probability weight of node j relative to node i

First, we calculate the point link strength and edge link strength of each pair of nodes according to the adjacency matrix. Then, the distance matrix is constructed. We calculate the density ρ_i and the shortest distance δ_i of each node based on the distance matrix. The linear fitting method is used to select the cluster centers. Finally, the DPOCD algorithm assigns the remaining nodes to the communities to which they belong based on their probability vectors. The flowchart of the DPOCD algorithm is shown in Fig. 2. Table I describes the notations used in the DPOCD algorithm.

A. Representation of Distance Between Nodes

The input for the proposed DPOCD algorithm differs from that for the DPC algorithm. The DPC algorithm takes the distance matrix D as input, but for complex networks, only the adjacency matrix A representing the relationship of links between nodes can be obtained. To fully exploit the advantages of DPC, it is first necessary to calculate distance matrix D. The simplest method is to define the distance between nodes as the reciprocal of the adjacency matrix value. However, the distribution of nodes in complex networks is sparse, especially in large-scale complex networks, and using this method will result in numerous infinite values in the calculated distance matrix. It also ignores the effect of the indirect link relationships between nodes on link strength, as shown in Fig. 3.

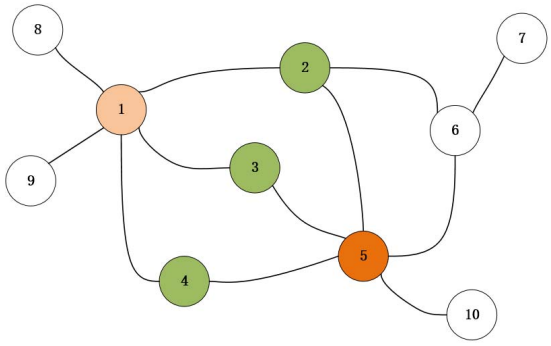


Fig. 3. Network structure diagram. Nodes 1 and 5 have three nodes in common. It can be seen that those two nodes are closely related. However, there is no direct link between them, the value of $A(1,5)$ is zero, and the distance between them is infinite, which is obviously unreasonable.

As shown in Fig. 3, although there is no direct link relationship between nodes 1 and 5, node 1 can be connected to node 5 through node 2. Therefore, the DPOCD algorithm considers nodes 1 and 5 to have an indirect link relationship. It defines the strength of edge links in the calculation of the distance matrix to indicate the influence of direct and indirect link relationships between nodes on the strength of the links. The strength of an edge link lst_{ij} is given by the following equation:

$$lst_{ij} = A_{ij} + \frac{r}{\min(w_i, w_j) + \eta} \sum_{t \in V_{ij}} A_{it} A_{tj} \tag{5}$$

where A denotes the adjacency matrix, V_{ij} denotes the set of nodes common to i and j, w_i denotes the total weight of node i, r denotes the range of the node weights, and η is a positive real number used to prevent the denominator from equaling zero.

We can also see from the figure that nodes 1 and 6 each have a direct link relationship with node 2, but the degrees of node 1 and node 6 differ, indicating that the strength of the link between nodes 1 and 2 is different from that between nodes 2 and 6. The DPOCD algorithm defines the influence of the degree of point link strength representation on the link strength between nodes. The point link strength vst_{ij} is given by the following equation:

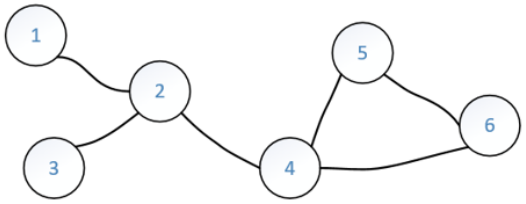
$$vst_{ij} = \frac{|V_{ij}| + 1}{\sqrt{\lambda |N_i^+| |N_j^+|}} \tag{6}$$

where N_i^+ denotes the set of nodes in node i and its neighborhood, V_{ij} denotes the set of nodes common to nodes i and j, and λ is a nonnegative real number.

Finally, the DPOCD algorithm calculates the matrix of distances between nodes based on the edge link strength and point link strength using (7). The results are shown in Fig. 4

$$\text{dist}_{ij} = \begin{cases} \dfrac{1}{lst_{ij} \times vst_{ij} + \varepsilon}, & i \neq j \\ 0, & i = j. \end{cases} \tag{7}$$

In (7), lst_{ij} denotes the edge link strength, vst_{ij} denotes the point link strength, and ε is a constant. The algorithm for calculating a distance matrix is presented in Algorithm 1.



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$w_1 = 1; w_2 = 3; w_3 = 1; w_4 = 3; w_5 = 2; w_6 = 2;\ r = \max(w_i) - \min(w_i) = 2.$

$$lst_{46} = A_{46} + \frac{r}{\min(w_4, w_6) + \eta} \sum_{t \in V_{46}} A_{4t} A_{t6} = 1.667$$

$$vst_{46} = \frac{|V_{46}| + 1}{\sqrt{\lambda |N_4^+| |N_6^+|}} = 1.826$$

$$dist_{46} = \frac{1}{lst_{46} * vst_{46} + \epsilon} = 0.247$$

Fig. 4. Calculation of distance between two nodes.

Algorithm 1 CalDisMx

Input: The adjacency matrix A; The number of nodes n;
Output: The distance matrix *dist*;

```
dist ← zeros(n)
for i ← 1 to n − 1 do
    for j ← i + 1 to n do
        V_ij ← commonNode(i, j)
        for each t ∈ V_ij do
            Calculate lst_ij via Equation (5)
        end for
        Calculate vst_ij via Equation(6)
        Calculate dist_ij via Equation(7)
        dist_ji ← dist_ij
    end for
end for
return dist
```

B. Calculation of ρ and δ

DPC uses ρ_i to denote the local density of node i, and the shortest distance δ_i is used to represent the distance between node i and the closest node having a higher density. Because the distribution of nodes in complex networks is sparse and the structure of communities is local in nature [23], the influence of global nodes on the local density of nodes should be considered when calculating the local density. Therefore, unlike the density peak clustering algorithm, the DPOCD algorithm only considers the influence of the k nearest neighbor nodes on the local density of nodes when calculating the local density. The definition of local density ρ_i is given by the following equation:

$$\rho_i = \sum_{j \in KNN_i} \exp\left(-\text{dist}_{ij}^2 / d_c^2\right) \tag{8}$$

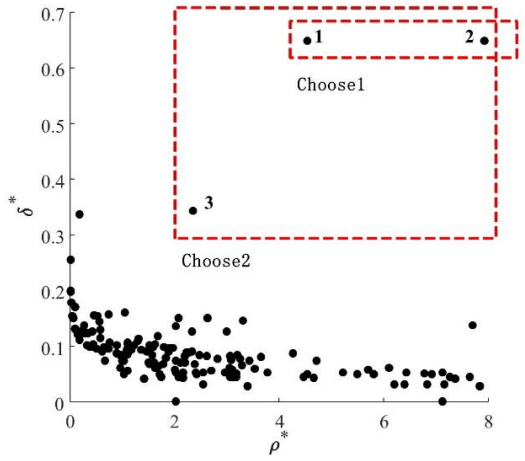


Fig. 5. ρ–δ decision diagram.

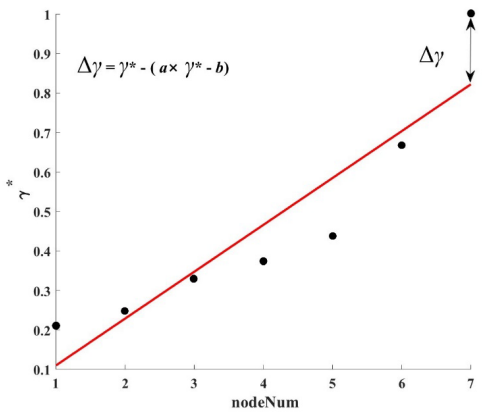


Fig. 6. γ^* ascending linear fit graph.

where KNN_i represents the k neighbors nearest to node I, and k is the average number of neighbors in the network. The shortest distance δ_i is given by (3).

For the effects of local density ρ and shortest distance δ on the selection of community centers to be consistent, ρ and δ must have the same range of values. Therefore, after calculating them, the DPOCD algorithm normalizes ρ and δ, as shown in the following equation:

$$\rho_i^* = \rho_i / \max\rho \tag{9}$$

$$\delta_i^* = \delta_i / \max\delta \tag{10}$$

where max ρ is the global maximum local density in the network and max δ is the global maximum shortest distance in the network. The algorithm for calculating the local density and shortest distance is given in Algorithm 2.

Algorithm 2 CalLocDenMinDis

Input: The distance matrix *dist*; The number of nodes n;
Output: Local density ρ^*; Minimum distance δ^*;

```
k ← average(∑_i Neigh(i))
for i ← 1 to n do
    for j ← 1 to k do
        Calculate ρ_i via Equation (8)
    end for
end for
for i ← 1 to n do
    for j ← 1 to n do
        if ρ_i = max ρ then
            δ_i ← max_j dist_ij
        else
            Calculate δ_i via Equation (3)
        end if
    end for
end for
Calculate ρ* via Equation (9)
Calculate δ* via Equation(10)
return ρ* and δ*
```

C. Adaptive Selection of Community Centers

To select the cluster centers, DPC draws a decision map with ρ as the horizontal coordinate and δ as the vertical coordinate and then manually selects nodes having large ρ and δ values as the cluster centers. This strategy reduces the efficiency of DPC and makes the selection result more subjective. As shown in Fig. 5, the horizontal coordinates of the decision diagram are ρ^* values, and the vertical coordinates are δ^* values. The nodes with small ρ^* and δ^* values are distributed in the lower left corner of the graph, whereas nodes 1 and 2 are located in the upper right corner of the graph; therefore, one option is to select nodes 1 and 2 as the cluster centers. However, node 3 also has large ρ^* and δ^* values relative to other nodes in the lower left corner of the decision diagram, and node 3 may also be a cluster center; therefore, another option is to select nodes 1, 2, and 3 as the cluster centers.

To improve the efficiency of the algorithm and avoid deterioration of the accuracy of the community center selection from subjective factors, the DPOCD algorithm incorporates linear fitting to create an adaptive method for community center selection. It is known from the DPC algorithm that only nodes with large values of ρ and δ in the decision graph are likely to be selected as cluster centers. Therefore, the DPOCD algorithm assumes that only nodes having both ρ^* and δ^* values greater than 80% of the average of the nodes are likely to be selected as community centers. To improve the efficiency of the algorithm, nodes that do not meet this condition are filtered out. For each node that meets the condition, the product γ of ρ^* and δ^* is calculated, as shown in the following equation:

$$\gamma = \rho^* \delta^*. \tag{11}$$

After γ has been calculated, the sorted γ is denoted by γ^*, and the index corresponding to γ is denoted by ind. The ascending sequence of γ^* is shown in Fig. 6, using the Karate Network dataset [36] as an example. The Karate Network is an abstraction of the membership and corresponding relationships of a Karate Club; it contains 34 nodes and 78 edges. After the screening for the eligible nodes has been completed as shown

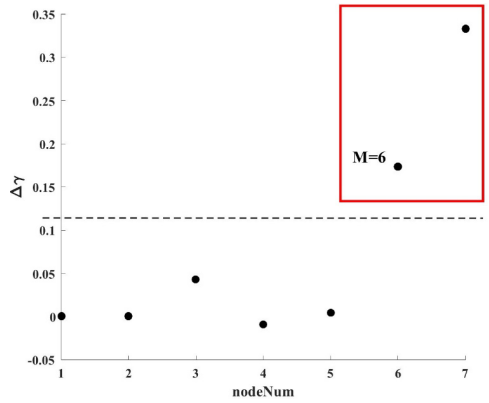


Fig. 7. Adaptive community center selection diagram.

in Fig. 6, the seven nodes remaining have the possibility of being selected as community centers in ascending order of γ^*.

Next, the estimated values corresponding to the γ^* for each node in the ascending γ^* plot are calculated from right to left using a linear fitting method. The estimated values of γ^* are denoted by γ', as shown in the following equation:

$$\gamma' = a \times \gamma^* + b \tag{12}$$

where a and b are the parameters obtained by linear fitting. As shown in Fig. 6, the DPOCD algorithm first fits a linear function from nodes 1 to 6 and the corresponding γ^*. The fit parameters are $a = 0.12$ and $b = -0.0085$. Then, the estimated value γ', from right to left, is calculated from the linear function, and so on. After completing the calculation of γ', the difference $\Delta\gamma$ is calculated as shown in the following equation:

$$\Delta\gamma = \gamma^* - \gamma' \tag{13}$$

$\Delta\gamma$ represents the difference between $\gamma*$ of a node and the corresponding estimated value γ'.

The corresponding $\Delta\gamma$ for the Karate Network dataset is shown in Fig. 7, where the dashed line corresponds to a scale computed by averaging $\Delta\gamma$ for all the remaining nodes. DPOCD records M as the smallest subscript of a node greater than the average $\Delta\gamma$ and adds a node with a node subscript not less than M to the set of candidate community centers, as shown in the following equation:

$$S = \{\text{ind}_t | M \le t \le |\text{ind}|\}. \tag{14}$$

As shown in Fig. 7, the smallest subscript for the node in the graph where $\Delta\gamma$ is greater than the average is 6; therefore, $M = 6$, and the corresponding nodes for nodes 6 and 7 are added to set S of candidate community centers.

Finally, because DPC assumes that the local density of a cluster center is higher than that of the nodes in its neighborhood and that it is a large distance from nodes having a higher local density, DPOCD considers that the local density and the shortest distance of the cluster center should be greater than the maximum local density and the shortest distance, respectively, of the k nearest neighbor nodes. If there are nodes in the candidate community center set S that do not meet these criteria, they are removed from S. The adaptive community center selection algorithm is given in Algorithm 3.

Algorithm 3 SelectCenter

Input: The set of local densities ρ^*; The set of minimum distance δ^*;
Output: The set of cluster centers S;
Calculate γ via Equation (11);
$\gamma^* \leftarrow$ arrange γ in ascending order;
$ind \leftarrow$ index vector of γ^* relative to γ;
for $i \leftarrow 1$ to $|ind|$ **do**
  Calculate γ' via Equation (12);
  Calculate $\Delta\gamma$ via Equation (13);
end for
for $i \leftarrow |ind|$ to 1 **do**
  if $\Delta\gamma_i < average(\Delta\gamma)$ **then**
    $M \leftarrow i + 1$; break;
  end if
end for
Identify candidate community center collections S via Equation (14)
for $i \leftarrow 1$ to $|S|$ **do**
  $KNN_i \leftarrow k$ nearest nodes to node i;
  for $j \leftarrow 1$ to k **do**
    if $\rho_i^* < \rho_j^*$ or $\delta_i^* < \delta_j^*$ **then**
      $S \leftarrow S - \{i\}$;
    end if
  end for
end for
return S

D. Allocation of Remaining Nodes

The residual node assignment mechanism of DPC assigns nodes to clusters that have higher local densities and are closest to it. This assignment mechanism results in nodes belonging to only one cluster. In a complex network, however, it is possible for nodes to belong to multiple communities; therefore, this assignment mechanism is unsuitable for detecting overlapping communities.

The residual node assignment mechanism of the proposed DPOCD algorithm, in contrast, has the capability of distinguishing overlapping nodes. To accomplish this, DPOCD defines a probability vector that indicates the probability of the node belonging to each community. For example, the probability vector for node i is expressed as $p_i = \{p_{i1}, p_{i2}, \ldots, p_{i|S|}\}$. If node i is the center of the community and the community is t, the probability vector p_i for node i is given by

$$p_{ij} = \begin{cases} 1, & j = t \\ 0, & j \ne t. \end{cases} \tag{15}$$

That is, community center i only belongs to community t. If node i is not the center of a community, its probability vector is determined by the k nodes that have a higher local density than node i and are closest to node i. The probability

vector p_i for such a node i is given by

$$p_i = \sum_{j \in cc_i} w_{ij} p_j \tag{16}$$

where cc_i denotes the set of k nodes with local density higher than node i and that are closest to node i, p_j denotes the probability vector for node j in cc_i, and w_{ij} denotes the probability weight of node j relative to node i. The probability weight w_{ij} is given by

$$w_{ij} = \frac{1}{\text{dist}_{ij} \sum_{m \in cc_i} (1/\text{dist}_{im})}. \tag{17}$$

After calculating the probability vectors for all nodes, DPOCD assigns the nodes to the communities to which they belong based on their probability vectors. The algorithm first assigns the node to the community corresponding to the maximum value in the probability vector. For example, in the probability vector of the node, if $\text{argmax}(p_i) = r$, node i will be assigned to community r. The algorithm then calculates the ratio of each of the other probabilities p_{iv} in the probability vector to the maximum probability p_{ir}. If p_{iv} satisfies $p_{iv}/p_{ir} \geq \zeta$, where ζ is a prespecified threshold, then node i is considered to be a node in community v at the same time. The procedure for allocating the remaining nodes is presented in Algorithm 4.

Algorithm 4 AllocateNodes

Input: The selected cluster centers S; The distance matrix *dist*; The number of nodes n;
Output: The node assignment matrix C;

```
C ← zeros(n, |S|)
for i ← 1 to |S| do
    Calculate p_i of S_i via Equation(15)
end for
for i ← 1 to n do
    if i ∉ S then
        for j ← 1 to k do
            Calculate w_ij via Equation(17)
            Calculate p_i via Equation(16)
        end for
    end if
end for
for i ← 1 to n do
    r ← argmax(p_i)
    C(i, r) ← 1
    for v ← 1 to |S| do
        if v ≠ r and p_iv / p_ir ≥ ζ then
            C(i, v) ← 1
        end if
    end for
end for
return C
```

E. Time Complexity Analysis

As described in Sections IV-A–D, the proposed DPOCD algorithm consists of four steps: calculation of the distance matrix, calculation of the local density and shortest distance, adaptive selection of community centers, and assignment of the remaining nodes. The time complexity for calculating the distance matrix (Algorithm 1) is $O(cn^2)$, where c represents the average number of common nodes and has a value that is much less than n. The time complexity for calculating the local density of nodes is $O(kn)$ and that for calculating the shortest distance of nodes is $O(n^2)$; therefore, the time complexity of Algorithm 2 is $O(kn + n^2)$. The time complexity for sorting γ in ascending order is $O(n \log n)$ and that for selecting a community center is $O(k|S|)$, where both k and $|S|$ are much less than n; therefore, the time complexity of Algorithm 3 is $O(n \log n + k|S|)$. The time complexity for calculating the probability vector is $O(kn)$ and that for assigning the remaining nodes is $O(|S|n)$; therefore, the time complexity of Algorithm 4 is $O(kn + |S|n)$. By combining the results of these analyses, the time complexity of the DPOCD algorithm is found to be $O(cn^2)$.

TABLE II
COMPARISON OF TIME COMPLEXITY FOR VARIOUS EXISTING METHODS WITH DPOCD

Algorithm	Time Complexity
OCDDP[30]	$O(nm+n^2log_2n)$
SLPA[17]	$O(tm)$
COPRA[16]	$O(vmlog(vm/n))$
LEMEX[23]	$O(n^2)$
MCMOEA[13]	$O(max\{N\times k^3_{max}, M^2\times PS, M\times PS\times gen_{max}\})$
DPOCD	$O(cn^2)$

Table II presents the time complexity of the proposed DPOCD and several existing overlapping community detection methods. The complexity of the OCDDP algorithm is $O(nm+n^2\log_2 n)$, where n is the number of nodes and m is the number of edges. The complexity of the SLPA algorithm is $O(tm)$, where t is predefined maximum number of iterations. For a sparse network in COPRA, the time complexity is $O(vm\log(vm/n))$, where v is the maximum number of communities in which a node can participate. The complexity of the LFMEX algorithm is $O(n^2)$, and the complexity of the MCMOEA algorithm is $O(\max\{N \times k^3_{\max}, M^2 \times \text{PS}, M \times \text{PS} \times \text{gen}_{\max}\})$, where M is usually smaller than or equal to n, PS is the size of population, and $\text{gen}_{\max}$ is the maximum number of generations. Finally, the complexity of our method is $O(cn^2)$, where c represents the average number of common nodes.

V. EXPERIMENTS

In this section, we will briefly introduce baseline algorithms, community validation metrics, and datasets. Then, we demonstrate and analyze our effectiveness evaluation on real-world networks and synthetic networks, and the experimental results are compared with the baseline algorithms.

A. Experiment Setup

To explore the feasibility of the DPOCD algorithm proposed in this article, this chapter compares the DPOCD

algorithm with five common overlapping community detection algorithms on synthetic network and real network datasets and evaluates the algorithm performance according to the evaluation index.

The hardware environment for all the experiments is as follows: Inter(R) Core(TM) i7-6700HQ CPU, 2.60 GHz, and 8 GB of memory. The DPOCD algorithm is implemented in MATLAB R2016a.

1) Baseline Algorithms: The baseline algorithms include:

OCDDP [30], an extension of DPC for overlapping community detection, requires human intervention to choose cluster centers.

SLPA [17], a speaker–listener-based information propagation algorithm. It has a threshold parameter r, and we set it as 0.05, for best results.

COPRA [16], an extension of label propagation algorithm in detecting overlapping communities in complex networks.

LFMEX [23], a method based on the local optimization of a fitness function, and it is an extension of LFM algorithm.

MCMOEA [13], a maximal clique-based MOEA for overlapping community detection.

2) Community Validation Metrics: In this study, two evaluation indexes of machine learning, *ONMI* [23] and the Rand index Ω [37], were used to measure the accuracy of the overlapping community detection algorithm.

To create a metric that can accurately evaluate the accuracy of an overlapping community detection algorithm in identifying communities, Lancichinetti *et al.* [23] extended *NMI* and proposed the index *ONMI*, given by

$$ONMI = 1 - \frac{1}{2}\left[\frac{H(X|Y)}{H(X)} + \frac{H(Y|X)}{H(Y)}\right] \tag{18}$$

where $H(X)$ denotes the entropy of random variable X, related to community division; $H(Y)$ denotes the entropy of random variable Y, related to standard division; and $H(X|Y)$ denotes the conditional entropy. $ONMI = 1$ indicates that the result of the community partition is the same as that of the standard partition, whereas $ONMI = 0$ indicates that the result of the community partition is completely different from that of the standard partition.

Collins and Dent [37] proposed the Rand index Ω for evaluating overlapping communities, given by

$$\Omega\ Index = \frac{N \cdot \sum_j \left|t_j(X) \cap t_j(Y)\right| - \sum_j \left|t_j(X)\right|\left|t_j(Y)\right|}{N^2 - \sum_j \left|t_j(X)\right|\left|t_j(Y)\right|} \tag{19}$$

where N denotes the total number of edges in a complex network, $t_j(X)$ denotes the set of pairs of nodes in community j that are divided by the algorithm partition X, and $t_j(Y)$ denotes the set of pairs of nodes in community j that are divided by the standard division Y.

In real networks, the traditional modularity function Q [38], [39] can only evaluate nonoverlapping communities. To measure the performance of an overlapping community detection algorithm, two extended modularity functions, EQ and Q_{ov}, are used to evaluate the quality of the overlapping communities detected by the algorithm.

TABLE III
PARAMETERS' INFORMATION OF THE SYNTHETIC NETWORK

maxk	40
Om	{2,3,4,5,6,7,8}
On	$0.1N$
mu	{0.2,0.3}

Shen *et al.* [9] proposed the EAGLE algorithm and introduced the extended modularity function EQ, given by

$$EQ = \frac{1}{2m}\sum_i \sum_{v,w \in C_i} \frac{1}{O_v O_w}\left[A_{vw} - \frac{k_v k_w}{2m}\right] \tag{20}$$

where m denotes the total number of edges in a complex network, C denotes the set of communities in the network, A represents the adjacency matrix, O_v denotes the number of communities to which node v belongs, and k_v denotes the degree of node v. Because (20) considers the number of communities to which the nodes belong, the EQ function can be used to evaluate the quality of overlapping communities.

To ensure the evaluation indexes more accurately reflect the quality of an overlapping community detection algorithm in identifying the community structure in a directed network, Nicosia *et al.* [10] considered the case of a directed graph, introduced the degree of entry and exit of nodes into the traditional modularity function Q, and proposed the extended modularity function Q_{ov}, given by

$$Q_{\text{ov}} = \frac{1}{m}\sum_{c \in C}\sum_{i,j \in V}\left[\beta_{l(i,j),c}A_{ij} - \frac{\beta^{\text{out}}_{l(i,j),c}k^{\text{out}}_i \beta^{\text{in}}_{l(i,j),c}k^{\text{in}}_j}{m}\right] \tag{21}$$

where m denotes the total number of edges in a complex network, C denotes the set of communities in the network, V denotes the set of nodes in the network, A denotes the adjacency matrix, $\beta_{l(i,j),c}$ indicates the contribution of edge $l(i,j)$ to the modularity of community c, k^{out}_i is the out-degree of node i, and k^{in}_j is the in-degree of node j.

Functions EQ and Q_{ov} are extensions of the modularity function Q in overlapping community detection, and their ranges are [0,1]. The larger the function value, the better the quality of the communities identified by the overlapping community detection algorithm and the stronger the ability of the algorithm to detect communities.

3) Test Suite of Datasets: In the synthetic network experiment, the LFR benchmark network is used as the dataset of the comparative experiment. The LFR benchmark network synthesizer provides a series of parameters to personalize the network topology. Among them, N is the number of nodes in the network, max c is the maximum size of the community, min c is the minimum size of the community, k is the average degree of nodes, max k is the maximum degree of nodes, *Om* is the number of communities to which overlapping nodes belong, *On* is the number of overlapping nodes in the network, and *mu* is a mixed parameter used to adjust the degree of difficulty in detecting community structure. The larger the *mu*, the more difficult it is for the algorithm to detect community structure correctly in the network. As shown in Table III, when synthesizing LFR benchmark networks, the average degree k

TABLE IV
REAL NETWORK DATASETS

Datasets	Nodes	Edges	Average Degree	Maximum Degree	Type
Karate	34	78	4.5882	17	Unweighted
Dolphins	62	159	5.1290	12	Unweighted
Polbooks	105	441	8.4000	25	Unweighted
Football	115	613	10.6609	12	Unweighted
Crime	829	1476	3.5537	25	Unweighted
Email	1133	5451	9.6222	71	Unweighted
Faa	1226	2615	3.9282	34	Unweighted
Propro	1870	2277	2.3561	56	Unweighted
Vidal	3133	6726	3.9253	129	Unweighted
Advogato	6541	51127	12.0119	803	Weighted
Hep-th	8361	15751	3.7677	50	Weighted
PGP	10680	24316	4.5536	205	Unweighted
Cond-mat	16726	47594	5.6910	107	Weighted
EAT	23219	325593	26.2662	1090	Weighted

of nodes in the network is set to 10, the maximum degree max k is set to 40, the minimum size min c of communities in the network is set to 20, the maximum size max c is set to 60, the number of nodes $N \in \{1000,5000,10\ 000\}$, the number of overlapping nodes On is set to $0.1N$, the mixed parameter $mu \in \{0.2,0.3\}$, and the number of communities to which the overlapping nodes belong $Om \in \{2,3,4,5,6,7,8\}$. Based on the above parameter settings, a total of 42 LFR benchmark networks are synthesized as datasets for comparison experiments. Then, the overlapping communities in the LFR benchmark network datasets are detected by the DPOCD algorithm and five other comparison algorithms proposed in this article. The accuracy of overlapping communities detected by the algorithms is evaluated by machine learning evaluation indexes Ω *Index* and *ONMI*.

To explore the performance of the proposed DPOCD algorithm in detecting overlapping communities in real networks, 14 real networks of various sizes were used as datasets for comparison experiments. As shown in Table IV, the smallest Karate network in the dataset consisted of 34 nodes and 78 edges, whereas the largest EAT network consisted of 23 219 nodes and 32 553 edges. The Advogato, Hep-th, Cond-mat, and EAT networks in the dataset shown in Table IV are weighted networks, whereas the others are not. The average and maximum degrees of nodes in Table IV reflect the complexity of the real network dataset.

B. Experiments and Analysis

Since synthetic networks have standard community partitions, and large real networks generally do not have standard community partitions, this section compares the performance of the proposed DPOCD algorithm with that of the five comparison algorithms OCDDP, SLPA, COPRA, LFMEX, and MCMOEA on synthetic networks and real network datasets, respectively.

1) Experiments on Synthetic Networks: Based on artificial synthetic network experiments, this section first analyses the performance of DPOCD algorithm. Fig. 8 shows the effect of the mixed parameter *mu* on the detection accuracy of DPOCD algorithm. As shown in Fig. 8, the detection accuracy of the DPOCD algorithm decreases with the increase in *Om*, because

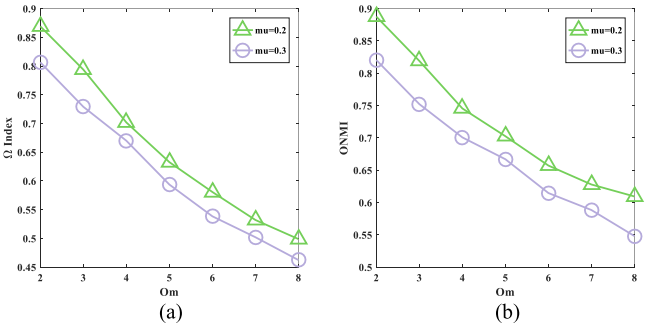


Fig. 8. Effect of mixed parameter *mu* on DPOCD algorithm performance. (a) and (b) Change in Ω *Index* and *ONMI* with *Om* when the number of nodes $N = 5000$ and the mixed parameter $mu \in \{0.2,0.3\}$, respectively. (a) Results of Ω *Index*. (b) Results of *ONMI*.

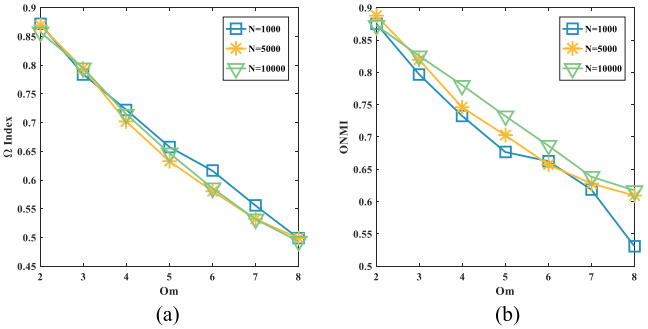


Fig. 9. Effect of the number of nodes N on the performance of the DPOCD algorithm. (a) and (b) When the mixed parameter $mu = 0.2$ and the number of nodes $N \in \{1000,5000,10\ 000\}$, the Ω *Index* and *ONMI* change with the number of communities *Om* belonging to the overlapping nodes. (a) Results of Ω *Index*. (b) Results of *ONMI*.

the increase in *Om* results in greater network overlap, and the algorithm is not easy to detect the correct community structure in the network. At the same time, the DPOCD algorithm performs better when $mu = 0.2$ than when $mu = 0.3$, because the structure of complex network becomes more complex with *mu*, and it is more difficult for the algorithm to detect overlapping communities in the network correctly. However, as shown in Fig. 8, when $mu = 0.3$, the recognition accuracy of the DPOCD algorithm does not differ much from that of $mu = 0.2$, which indicates that the DPOCD algorithm has strong robustness.

Fig. 9 discusses the effect of the number of nodes N on the detection accuracy of the DPOCD algorithm. As shown in Fig. 9, the Ω *Index* and *ONMI* corresponding to the algorithm decrease with the increase in *Om* in three complex networks of different sizes because the increase in network overlap makes it more difficult for the algorithm to detect overlapping communities. At the same time, as shown in Fig. 9(a), the difference of Ω *Index* corresponding to the DPOCD algorithm in three different sizes of networks is small, indicating that the DPOCD algorithm is suitable for all sizes of networks, and the performance of the algorithm does not change significantly with the change in network size. As shown in Fig. 9(b), the *ONMI* corresponding to the DPOCD algorithm at $N = 10\ 000$ is larger than that at $N \in \{1000,5000\}$, indicating that the

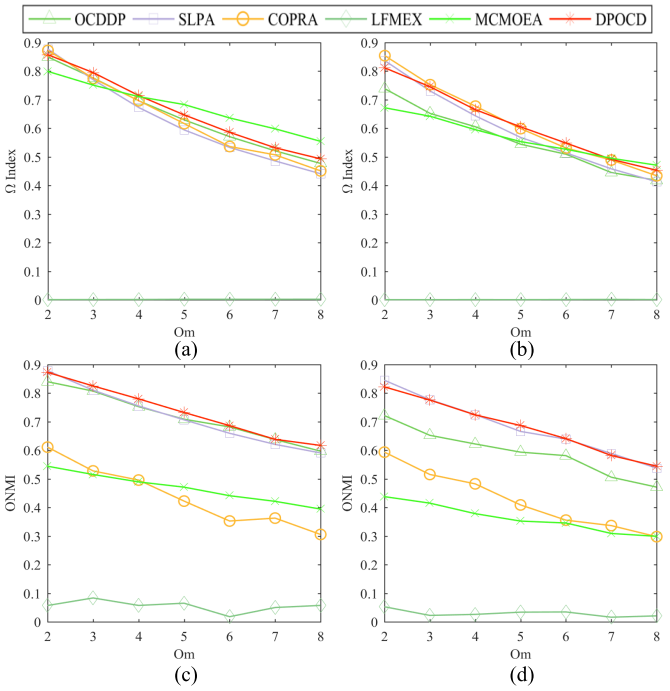


Fig. 10. Comparison results under the influence of mixed parameter *mu*. (a)–(d) Results of the DPOCD algorithm and the five comparison algorithms. **Ω** *Index* and *ONMI* change with the number of overlapping nodes *Om*, when the number of nodes $N = 10\,000$ and the mixed parameter $mu \in \{0.2,0.3\}$, respectively. (a) $mu = 0.2$. (b) $mu = 0.3$. (c) $mu = 0.2$. (d) $mu = 0.3$.

DPOCD algorithm performs better in detecting overlapping community structures in large complex networks.

Next, this section compares DPOCD with five other algorithms based on artificial synthetic network experiments. The comparison results of the algorithm under the influence of the mixed parameter *mu* are shown in Fig. 10. As shown in Fig. 10, the detection accuracy of the DPOCD algorithm and the OCDDP, SLPA, COPRA, and MCMOEA algorithms decreases with increasing *Om*. The LFMEX algorithm does not show a significant downward trend owing to the trend of zero detection accuracy. At the same time, as shown in Fig. 10, the **Ω** *Index* and *ONMI* corresponding to the proposed DPOCD algorithm are higher than the other five comparison algorithms on the whole, indicating that the DPOCD algorithm is better than the other five algorithms.

Fig. 11 shows the effect of node number *N* on the detection accuracy of the DPOCD algorithm and other five comparison algorithms. As can be seen, the **Ω** *Index* and *ONMI* of the DPOCD algorithm are better than those of the other five comparison algorithms in the network of various sizes. Furthermore, the **Ω** *Index* and *ONMI* of the algorithm decrease steadily with increasing *Om*, which again verifies the robustness of the proposed DPOCD algorithm.

2) Experiments on Real Social Networks: In the real network experiment, the modularity functions *EQ* and Q_{ov} are used as the evaluation index of overlapping community detection algorithm to quantitatively evaluate the quality of the community structure identified in 14 real network datasets of various sizes by the DPOCD algorithm and other five comparison algorithms proposed in this article. Table V shows

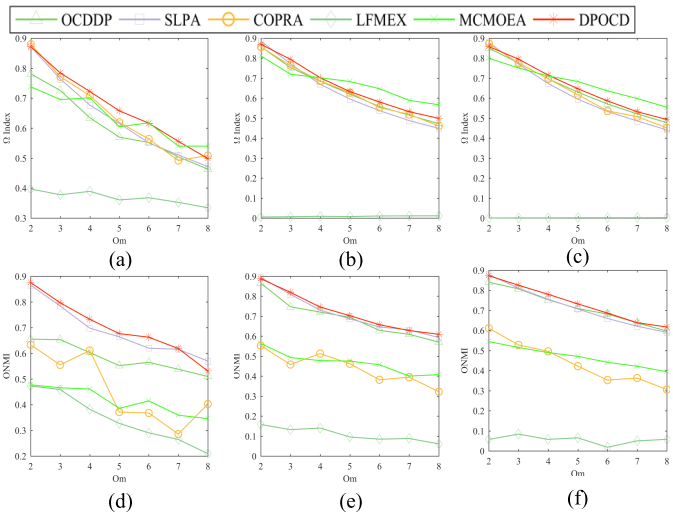


Fig. 11. Comparison results under the influence of number of nodes *N*. As shown in (a)–(c), when $N = 1000$, the LFMEX algorithm has a low accuracy. When $N \in \{5000,10\ 000\}$, the experimental results indicate that LFMEX is not suitable for large networks. Meanwhile, as shown in (d), the detection accuracy of COPRA varies significantly with changing *Om*, which indicates that COPRA has poor robustness. (a) $N = 1000$. (b) $N = 5000$. (c) $N = 10\ 000$. (d) $N = 1000$. (e) $N = 5000$. (f) $N = 10\ 000$.

TABLE V

EQ VALUE COMPARISON OF ALGORITHMS IN REAL NETWORKS

Datasets	OCDDP	SLPA	COPRA	LFMEX	MCMOEA	DPOCD
Karate	**0.3833**	0.3607	0.2387	0.3239	0.1649	0.3718
Dolphins	0.4992	0.5021	0.3799	0.4437	0.2493	**0.5051**
Polbooks	0.5008	0.4949	0.4501	**0.5206**	0.2375	0.4986
Football	0.5157	**0.5948**	0.5910	0.5179	0.4609	0.5526
Crime	0.4240	0.2140	0.0174	0.4571	0.2973	**0.4875**
Email	0.4719	0.4186	0.0104	0.3768	0.2088	**0.5089**
Faa	0.4156	0.2408	0.3981	0.4325	0.2777	**0.4984**
Propro	0.6652	0.5368	0.1446	0.4996	0.5863	**0.6869**
Vidal	0.4364	0.3834	0.1611	0.2538	0.2916	**0.5153**
Advogato	0.2421	0.1998	0.1656	0.1685	-	**0.2961**
Hep-th	0.6537	0.6079	0.3126	0.1434	0.5684	**0.6678**
PGP	0.6062	0.7063	0.6210	0.1130	0.5123	**0.7577**
Cond-mat	0.6583	0.6653	0.4788	0.0667	0.5792	**0.6951**
EAT	0.0587	0.0587	0.0166	0.0609	-	**0.1374**

TABLE VI

Q_{ov} VALUE COMPARISON OF ALGORITHMS IN REAL NETWORKS

Datasets	OCDDP	SLPA	COPRA	LFMEX	MCMOEA	DPOCD
Karate	0.6618	0.6904	0.7026	0.7021	0.6485	**0.7543**
Dolphins	0.7419	0.7468	0.7383	0.7047	0.2622	**0.7474**
Polbooks	0.8266	0.8420	0.8342	0.8029	0.2837	**0.8465**
Football	0.6044	0.7087	0.7069	0.6781	0.5336	**0.7231**
Crime	0.4301	0.2200	0.0197	**0.6150**	0.1877	0.4971
Email	0.5770	**0.6532**	0.0380	0.5494	0.1373	0.6177
Faa	0.4560	0.2703	0.5557	**0.6525**	0.2125	0.5706
Propro	0.6745	0.5465	0.1412	**0.7157**	0.5359	0.7085
Vidal	0.4342	0.3955	0.4168	0.1535	0.2023	**0.5475**
Advogato	0.3020	0.1520	0.0598	0.0166	-	**0.5131**
Hep-th	0.6558	0.6118	0.3141	0.2729	0.5233	**0.7018**
PGP	0.6108	0.7254	0.6514	0.2175	0.4559	**0.7737**
Cond-mat	0.6579	0.6678	0.4805	0.1456	0.4918	**0.7016**
EAT	0.0141	0.0000	0.0173	0.0632	-	**0.1784**

the algorithm comparison results with *EQ* as the evaluation index, and Table VI shows the algorithm comparison results with Q_{ov} as the evaluation index. Since there are outliers in the Advogato and EAT datasets, the MCMOEA algorithm is not suitable to be tested with these two datasets.

From Table V, we can see that the proposed DPOCD algorithm performs well in both weighted and unweighted networks. The *EQ* value of the DPOCD algorithm occupies

11 best results among 14 real network datasets of various sizes, which shows that the performance of the DPOCD algorithm in detecting overlapping communities is better than the other five comparison algorithms. At the same time, 9 of the 11 best results correspond to network datasets with node sizes larger than 1000, so the DPOCD algorithm performs better in detecting overlapping community structures in large network datasets. From Table V, we can see that the OCDDP algorithm performs better than the DPOCD algorithm in the Karate and Polbooks datasets, and its *EQ* value is 1.15% and 0.22% higher than the DPOCD algorithm, respectively. However, the *EQ* value of the DPOCD algorithm is 8.28%, 7.89%, 15.15%, and 7.87% higher than the OCDDP algorithm in the Faa, Vidal, PGP, and EAT datasets, respectively. Meanwhile, the *EQ* value of the SLPA algorithm in the Football dataset is 4.22% higher than that of the DPOCD algorithm, but the *EQ* value of the DPOCD algorithm in the Faa, Propro, Vidal, and Advogato datasets is 25.76%, 15.01%, 13.19%, and 9.63% higher than that of the SLPA algorithm, respectively. In addition, the *EQ* value of the LFMEX algorithm in the Polbooks dataset is 2.20% higher than that of the DPOCD algorithm; however, the *EQ* value of the DPOCD algorithm is 26.15%, 52.44%, 64.47%, and 62.84% higher than that of the LFMEX algorithm in the Vidal, Hep-th, PGP, and Cond-mat datasets, respectively, and it can be seen that the overall performance of the DPOCD algorithm is better than the OCDDP, SLPA, and LFMEX algorithms.

Table VI shows the evaluation results of the modularity function Q_{ov} on the community structure detected by the DPOCD algorithm and other five comparison algorithms in 14 real network datasets of various sizes. From Table VI, we can see that the Q_{ov} value of the DPOCD algorithm occupies ten best results among 14 real network datasets of various sizes, which further verifies the advantages of the DPOCD algorithm. In Table VI, the Q_{ov} value of the SLPA algorithm in the Email dataset is 3.55% higher than that of the DPOCD algorithm; however, the Q_{ov} value of the DPOCD algorithm is 27.71%, 30.03%, 36.11%, and 17.84% higher than that of the SLPA algorithm in the Crime, Faa, Advogato, and EAT datasets, respectively. The Q_{ov} values of the LFMEX algorithm in the Crime, Faa, and Propro datasets are 11.79%, 8.19%, and 0.72% higher than those of the DPOCD algorithm, but the Q_{ov} values of the DPOCD algorithm in the Advogato, Hep-th, PGP and Cond-mat datasets are 49.65%, 42.89%, 55.62%, and 55.60% higher than the LFMEX algorithm, respectively. The above results further verify that the overall performance of the DPOCD algorithm is better than the SLPA and LFMEX algorithms.

Compared with five overlapping community detection algorithms, OCDDP, SLPA, COPRA, LFMEX, and MCMOEA, we can see that the DPOCD algorithm proposed in this article is very robust. It is superior to the comparison algorithm and performs better in large-scale complex networks.

Other algorithms based on link strength, such as the algorithm OCDDP, are proposed in [30]. In terms of time complexity, the overall complexity of OCDDP is $O(nm+n^2\log_2 n)$, and the overall time complexity of our algorithm is $O(cn^2)$, where n represents the number of nodes, and c represents the average number of common nodes, which is much less than n. In the experimental part, our algorithm has higher *EQ* and Q_{ov} values than the OCDDP algorithm on most comparison datasets. Therefore, we can conclude that our algorithm has a better performance.

The *EQ* values obtained by our algorithm on three large-scale datasets (PGP, Cond-mat, and EAT) are 0.7577, 0.6951, and 0.1374 respectively, and the Q_{ov} values are 0.7737, 0.7016, and 0.1784, respectively. The results show that the *EQ* values and Q_{ov} values obtained by our algorithm are higher than those obtained by the comparison algorithm. Therefore, it can be said that our algorithm has a better performance on large-scale datasets.

VI. Conclusion

In this work, we propose an overlapping community detection (DPOCD) algorithm as an extension of DPC. The proposed algorithm first defines edge link strength and point link strength and calculates the distance matrix. Then, it selects the clustering centers adaptively and uses the linear fitting method. Finally, it defines probability vector to represent the degree of belonging of nodes to each community and allocates the remaining nodes. Experiments on synthetic and real networks show that our method is effective and robust, which performs better than those methods compared in the experiments on large-scale complex networks. In the future, this method will be further explored for enhancing the efficiency for large-scale complex networks. An interesting research is to optimize the flow of our algorithm through the parallel processing mechanism.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions that helped them to improve the presentation of this article considerably.

References

[1] J. Galaskiewicz, "The structure of community organizational networks," *Social Forces*, vol. 57, no. 4, p. 1346, Jun. 1979.
[2] M. E. J. Newman, *Networks: An Introduction*. Oxford, U.K.: Oxford Univ. Press, 2010.
[3] S. Bahadori, H. Zare, and P. Moradi, "PODCD: Probabilistic overlapping dynamic community detection," *Exp. Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114650.
[4] Y. Gao, X. Yu, and H. Zhang, "Overlapping community detection by constrained personalized PageRank," *Exp. Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114682.
[5] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.
[6] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[7] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Portland, OR, USA, 1996, pp. 226–231.
[8] S. Maity and S. K. Rath, "Extended clique percolation method to detect overlapping community structure," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Delhi, India, Sep. 2014, pp. 31–37.
[9] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Phys. A, Stat. Mech. Appl.*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009.
[10] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," *J. Stat. Mech., Theory Exp.*, vol. 2009, no. 3, Mar. 2009, Art. no. P03024.

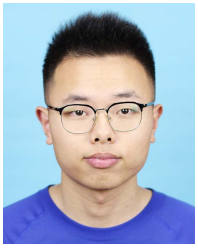
[11] X. Qian, L. Yang, and J. Fang, "Overlapping community detection based on community connection similarity of maximum clique," in *Proc. Int. Comput. Frontier Conf.*, Beijing, China, 2018, pp. 241–252.
[12] M. Esmaeili, H. M. Saad, and A. Nosratinia, "Semidefinite programming for community detection with side information," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1957–1973, Apr. 2021.
[13] X. Wen *et al.*, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 363–377, Jun. 2017.
[14] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, "A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2703–2716, Sep. 2017.
[15] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, pp. 1–11, Sep. 2007.
[16] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, pp. 2011–2024, 2009.
[17] J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks," in *Proc. 16th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining*, Kuala Lumpur, Malaysia, 2012, pp. 25–36.
[18] E. Alghamdi and D. Greene, "Semi-supervised overlapping community finding based on label propagation with pairwise constraints," in *Proc. Int. Conf. Complex Netw. Appl.*, London, U.K., 2018, pp. 316–327.
[19] M. Lu, Z. Zhang, Z. Qu, and Y. Kang, "LPANNI: Overlapping community detection using label propagation in large-scale complex networks," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1736–1749, Sep. 2019.
[20] B. Huang, C. Wang, and B. Wang, "NMLPA: Uncovering overlapping communities in attributed networks via a multi-label propagation approach," *Sensors*, vol. 19, no. 2, p. 260, Jan. 2019.
[21] R. Gao, S. Li, X. Shi, Y. Liang, and D. Xu, "Overlapping community detection based on membership degree propagation," *Entropy*, vol. 23, no. 1, p. 15, Dec. 2020.
[22] C. Wang, W. Tang, B. Sun, J. Fang, and Y. Wang, "Review on community detection algorithms in social networks," in *Proc. IEEE Int. Conf. Prog. Informat. Comput. (PIC)*, Nanjing, China, Dec. 2015, pp. 551–555.
[23] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, Mar. 2009, Art. no. 033015.
[24] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using neighborhood-inflated seed expansion," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1272–1284, May 2016.
[25] Z.-J. Jian, H.-S. Ma, and J.-W. Huang, "CLOSE: Local community detection by local structure expansion in a complex network," in *Proc. Int. Conf. Technol. Appl. Artif. Intell. (TAAI)*, Kaohsiung, Taiwan, Nov. 2019, pp. 1–6.
[26] H. Liu, L. Fen, J. Jian, and L. Chen, "Overlapping community discovery algorithm based on hierarchical agglomerative clustering," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 3, Mar. 2018, Art. no. 1850008.
[27] Z. Yu, J. Chen, K. Quo, Y. Chen, and Q. Xu, "Overlapping community detection based on random walk and seeds extension," in *Proc. 12th Chin. Conf. Comput. Supported Cooperat. Work Social Comput.*, Chongqing, China, Sep. 2017, pp. 18–24.
[28] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Trans. Computa. Social Syst.*, vol. 5, no. 4, pp. 1021–1033, Dec. 2018, doi: 10.1109/TCSS.2018.2879494.
[29] F. Cheng, C. Wang, X. Zhang, and Y. Yang, "A local-neighborhood information based overlapping community detection algorithm for large-scale complex networks," *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 543–556, Apr. 2021.
[30] X. Bai, P. Yang, and X. Shi, "An overlapping community detection algorithm based on density peaks," *Neurocomputing*, vol. 226, pp. 7–15, Feb. 2017.
[31] M. Xu, Y. Li, R. Li, F. Zou, and X. Gu, "EADP: An extended adaptive density peaks clustering for overlapping community detection in social networks," *Neurocomputing*, vol. 337, pp. 287–302, Apr. 2019.
[32] H. Lu, Z. Shen, X. Sang, Q. Zhao, and J. Lu, "Community detection method using improved density peak clustering and nonnegative matrix factorization," *Neurocomputing*, vol. 415, pp. 247–257, Nov. 2020.
[33] X.-F. Wang and Y. Xu, "Fast clustering using adaptive density peak detection," *Stat. Methods Med. Res.*, vol. 26, no. 6, pp. 2800–2811, Dec. 2017.
[34] J. Jiang, X. Tao, and K. Li, "DFC: Density fragment clustering without peaks," *J. Intell. Fuzzy Syst.*, vol. 34, no. 1, pp. 525–536, Jan. 2018.
[35] M. Wang, W. Zuo, and Y. Wang, "An improved density peaks-based clustering method for social circle discovery in social networks," *Neurocomputing*, vol. 179, pp. 219–227, Feb. 2016.
[36] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.
[37] L. M. Collins and C. W. Dent, "Omega: A general formulation of the Rand index of cluster recovery suitable for non-disjoint solutions," *Multivariate Behav. Res.*, vol. 23, no. 2, pp. 231–242, Apr. 1988.
[38] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, pp. 1–16, Feb. 2004.
[39] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

Liping Sun received the Ph.D. degree from Anhui Normal University, Wuhu, China, in 2015.

She is currently a Professor with the School of Computer and Information, Anhui Normal University. Her current research interests include data mining and intelligent computing.

Tao Ye received the B.Sc. degree from Anhui Normal University, Wuhu, China, in 2020, where he is currently pursuing the master's degree with the School of Computer and Information.

His current research interests include complex network and data mining.



Jian Sun received the B.Sc. degree from Anhui Normal University, Wuhu, China, in 2020. He is currently pursuing the master's degree with Software Institute, Nanjing University, Nanjing, China.

His current research interests include complex network and data mining.

Xiaoyu Duan received the B.Sc. degree from Anhui University of technology, Ma'anshan, China, in 2019. He is currently pursuing the master's degree with the School of Computer and Information, Anhui Normal University, Wuhu, China.

His current research interests include complex network and data mining.

Yonglong Luo received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, 2005.

He is currently a Professor with the School of Computer and Information, Anhui Normal University, Wuhu, China. His current research interests include information security and spatial data processing.