Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 3790590, 15 pages
http://dx.doi.org/10.1155/2016/3790590

Research Article

A Neighborhood-Impact Based Community Detection Algorithm via Discrete PSO

Dongqing Zhou and Xing Wang

Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China

Correspondence should be addressed to Dongqing Zhou; qq_eastz@126.com

Received 30 August 2015; Revised 6 December 2015; Accepted 31 December 2015

Academic Editor: Daniel Aloise

The paper addresses particle swarm optimization (PSO) into community detection problem, and an algorithm based on new label strategy is proposed. In contrast with other label propagation strategies, the main contribution of this paper is to design the definition of the impact of node and take it into use. Special initialization and update approaches based on it are designed in order to make full use of it. Experiments on synthetic and real-life networks show the effectiveness of proposed strategy. Furthermore, this strategy is extended to signed networks, and the corresponding objective function which is called modularity density is modified to be used in signed networks. Experiments on real-life networks also demonstrate that it is an efficacious way to solve community detection problem.

1. Introduction

Complex networks have attracted increasing attention of researchers from different fields, such as physics, sociology, mathematics, and computer science [1]. The related theory has been widely applied in many aspects, including the Internet [2], communication [3], biology [4], and economy [5]. Networks are usually composed of subgroup structures, whose interconnections are dense and intraconnections are sparse. This property is called community structure. Detecting community structure is one of the fundamental issues in networks study; it could reveal latent meaningful structure in networks. It is particularly important to detect the structure of the commonly used networks, such as daily social networks, recommendation system, and nation power distribution networks [6].

Community detection is an NP-hard problem [7]; traditional methods for detecting communities in networks can be concluded in two categories: graph partitioning and hierarchical clustering. The graph partitioning method has been widely used in computer science and the related fields [8]. However, this method needs to know the number of communities and the size of them before partitioning. The hierarchical clustering methods include agglomerative clustering algorithm and divisive clustering algorithm; they do not require the number or size of the communities [9, 10]. However, the results of this method depend on the specific similarity measure adopted.

In recent years, a lot of effort has been made to develop new approaches and algorithms to detect and quantify community structure in complex networks. Newman introduces the modularity as a stopping criterion originally, which becomes one of the most commonly used and best known quality functions. A lot of modularity based methods have been proposed, including modularity optimization, simulated annealing [11], extremal optimization [12], and spectral optimization [13]. These modularity based methods provide an outstanding way to solve the community problem and many researches are carried out based on these methods. Pizzuti models the community problems as single-objective optimization problem and multiobjective optimization problem, respectively, in [14] and [15]. Arenas and Díaz-Guilera investigate the connection between the dynamics of synchronization and the modularity on complex networks in [16].

However, the modularity has the disadvantage of resolution limits found by Fortunato and Barthélemy in [17]. It is that the modularity has an intrinsic scale and the modularity may not be resolved even in the extreme case if it is smaller than this scale.

Li et al. propose a quantitative measure called the modularity density, which uses the average modularity degree to evaluate the partition of a network in [18]. This method provides a mesoscopic way to describe the network structure and overcomes the resolution limit of modularity. The larger the value of modularity density is, the more accurate a partition is. Thus, the community detection can be viewed as an optimization problem of finding a partition of a network which has the maximum modularity density.

Particle swarm optimization (PSO) algorithm is successfully used in many optimization fields [19–22]. It was proposed to solve continuous optimization problems at first; then Kennedy and Eberhart develop the continuous PSO to discrete binary PSO in [23]. Chen et al. propose the candidate solution and velocity based on discrete PSO in [24]. Recently, Gong et al. use discrete PSO to solve the community detection problem in [25]. Compared with traditional optimization methods, the discrete PSO is simple to implement and it has a high speed of convergence. It needs to know neither the number of the communities nor the size of the communities. What is more, bare mathematical assumptions that may be needed in the conventional methods are ignored. These advantages make it feasible and promising in solving community detection problems.

In this paper, a new algorithm based on discrete PSO is proposed to solve community detection problems by optimizing the modularity density function. We design a definition called the impact of node in this paper, which takes the neighborhood information into consideration. A new initialization and updating strategy based on the impact of node is proposed. At last, we modify the modularity density function to signed networks according to the character of community, and the proposed strategies are also extended to signed networks.

The rest of this paper is organized as follows. Section 2 is a review of the related introduction of modularity density. Section 3 gives a detailed description of the proposed method. In Section 4, the experimental results of the proposed algorithm in comparison with other approaches are shown. At last the conclusions are summarized in Section 5.

2. Community Structure and Modularity Density

A network can be modeled as $G = (V, E)$, where V and E represent the vertices and edges, respectively. Assume that $\mathbf{A}$ is the adjacency matrix of G. If there is a link between nodes i and j, $A_{ij} = 1$; otherwise $A_{ij} = 0$. Suppose that S is subgraph which belongs to G, and i is a node which belongs to S. k_i is the degree of the node i; $k_i^{\text{in}} = \sum_{i,j\in S} A_{ij}$, $k_i^{\text{out}} = \sum_{i\in,j\notin S} A_{ij}$. The community of a network usually has the following property:

$$\sum_{i\in S} k_i^{\text{in}} > \sum_{i\in S} k_i^{\text{out}}. \tag{1}$$

It means that the sum of all degrees within the community is larger than the sum of all degrees toward the rest of the network.

If the links between different nodes have negative or positive signs, they are called the signed networks. The signed networks can be modeled as $G = (V, \text{PE}, \text{NE})$. V is the set of nodes; PE and NE are the positive and negative links, respectively. Let e_{ij} be the link between nodes i and j; then the adjacency matrix $\mathbf{A}$ can be formulated as follows: if $e_{ij} \in \text{PE}$, $A_{ij} = 1$; if $e_{ij} \in \text{NE}$, $A_{ij} = -1$. Then the signed networks usually have the following property:

$$\begin{gathered} \sum_{i\in S} \left(k_i^{+}\right)^{\text{in}} > \sum_{i\in S} \left(k_i^{+}\right)^{\text{out}}, \\ \sum_{i\in S} \left(k_i^{-}\right)^{\text{in}} < \sum_{i\in S} \left(k_i^{-}\right)^{\text{out}} \end{gathered} \tag{2}$$

in which

$$\begin{gathered} \left(k_i^{+}\right)^{\text{in}} = \sum_{i,j\in S, e_{ij}\in \text{PE}} A_{ij}, \\ \left(k_i^{-}\right)^{\text{in}} = \sum_{i,j\in S, e_{ij}\in \text{NE}} \left|A_{ij}\right|, \\ \left(k_i^{+}\right)^{\text{out}} = \sum_{j\in S, e_{ij}\in \text{PE}} A_{ij}, \\ \left(k_i^{-}\right)^{\text{out}} = \sum_{j\in S, e_{ij}\in \text{NE}} \left|A_{ij}\right|. \end{gathered} \tag{3}$$

In an intuitive way, take the real-life networks SPP and GGS as examples (please refer to Section 4.5) for signed networks; the solid line and dashed line represent positive and negative links, respectively. The internal positive degree in a community is dense, and the negative external degree between different communities is dense.

Community detection can be formulated as a *modularity* (Q) optimization problem. Q was proposed by Newman in [13], which aims at finding a partition of the network. Suppose that there is a graph G' whose edges are drawn at random; it has the same distribution of degrees as G. Modularity is such a measurement that maximizes the sum of the inner edges over all the modules of G minus that of G' which has the expected sum of number of inner edges [26]. As Q is an evaluation function to estimate the community structure of the network, the larger the value of Q is, the clearer the structure of the community is. Otherwise, the structure is more obscure. The mathematical description of Q is as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(i, j), \tag{4}$$

where m is the number of edges in the network. If i and j are in the same community, $\delta(i, j) = 1$. Otherwise, $\delta(i, j) = 0$. According to [18], another form of Q is as follows:

$$Q = \sum_{i=1}^{N} \left[\frac{L(V_i, V_i)}{L(V, V)} - \left(\frac{L(V_i, V)}{L(V, V)} \right)^2 \right], \tag{5}$$

where N is the number of communities and $L(V_k, V_n) = \sum_{i \in V_k, j \in V_n} A_{ij}$, $L(V, V) = 2m$, and $L(V_i, V) = \sum_{j \in V_i, k \in V} A_{jk}$.

A class of methods aiming to maximize the modularity has been developed. However, the modularity has the disadvantage of resolution limits that it contains an intrinsic scale which depends on the size of links in the network. It cannot detect them exactly when the modules are smaller than this scale.

Modularity density (D) was proposed by Li et al. in [18] to evaluate the partition of a network based on the concept of average modularity degree. It overcomes the resolution limit in community detection:

$$D = \sum_{i=1}^{N} \frac{L(V_i, V_i) - L(V_i, \overline{V_i})}{|V_i|}. \tag{6}$$

$L(V_i, V_i)/|V_i|$ and $L(V_i, \overline{V_i})/|V_i|$ mean the average internal and external degrees of the ith community, respectively. D tries to maximize the average internal degree and minimize the average external degree of the communities. D is related to the density of subgraphs; it provides a way to overcome the problem that Q is sensitive to the the size of network and interconnections of modules. Thus, we could use D to decide whether the networks are partitioned into correct communities. According to the definition of modularity density D, the larger the value of D is, the more accurate a partition is. Then Li et al. improved D to a general version by setting a parameter λ to the proportion of average internal degree and external degree:

$$D_\lambda = \sum_{i=1}^{N} \frac{2\lambda L(V_i, V_i) - 2(1-\lambda) L(V_i, \overline{V_i})}{|V_i|}. \tag{7}$$

D_λ is a convex combination of ratio cut and ratio association. It tries to maximize the density of links inside a community and minimize the density of links among different communities. When $\lambda = 1$, D_λ is equal to ratio association; when $\lambda = 0$, D_λ is equal to ratio cut; when $\lambda = 0.5$, D_λ is equal to D. We can decompose the network into large communities when using small λ; otherwise, small communities are obtained. And for this, more details and levels of the network can be found. In this paper we use the discrete PSO algorithm to optimize D_λ to detect community structure.

3. The Description of Proposed Algorithm

In this section, the proposed algorithm for community detection is described in detail. First, the objective function of the algorithm is given. Next, the impact of node is described. Meanwhile, the particle swarm initialization and updating are presented. At last, the framework of the proposed algorithm is elaborated, and the complexity analysis is presented.

3.1. Objective Function. In this paper, we adopt D_λ as the objective function because of its efficiency in detecting community structure, which has been described in detail in Section 2. In order to solve signed network problem, we extend it to (8). It is consistent with the character of signed networks:

$$\begin{aligned} D_\lambda^+ &= \sum_{i=1}^{N} \frac{2\lambda L(V_i, V_i)^+ - 2(1-\lambda) L^+(V_i, \overline{V_i})}{|V_i|}, \\ D_\lambda^- &= -\sum_{i=1}^{N} \frac{2\lambda L(V_i, V_i)^- - 2(1-\lambda) L^-(V_i, \overline{V_i})}{|V_i|}, \\ D_\lambda &= D_\lambda^+ + D_\lambda^-. \end{aligned} \tag{8}$$

3.2. The Impact of Node. In order to solve community detection problem with discrete PSO, label propagation is introduced in [25]. The label of node is used to assign the node to different communities, and the nodes which have the same label are considered to be in the same community. This label propagation strategy considers the number of nodes with the same label in the neighborhood to update the current node's label. For example, consider a node i whose neighbors are $i^1, i^2, \ldots, i^n$, when using this label propagation to update the label of node i, check all the labels of $i^1, i^2, \ldots, i^n$ to find the label which appears the most, and then node i is assigned to this label. But actually, the contribution of each node to the neighborhood is not the same as the node we choose. In this subsection, a new definition that defines the "impact of node" is introduced and a new label propagation based on the "impact of node" is proposed. Consider a node i whose neighbors are $i^1, i^2, \ldots, i^k$; then the impact of node i^j on node i can be defined as follows:

$$\text{ion}(i^j) = \text{degree}(i^j) * \text{num}(l(i^j)), \tag{9}$$

where ion is the impact of node, degree (i^j) is the degree of node i^j, $l(i^j)$ means the label of node i^j, and num $(l(i^j))$ refers to the number of neighbors of i that have the same label with node i^j. It is noticed that the impact of node is usually different when the node we choose is in the neighborhood of different nodes. For signed networks, we only consider the positive links for that the number of negative links in a community is usually less than that of negative links between communities, and two nodes connected with negative links are usually located in different communities.

The ion describes the effectiveness of the node in the network, and the value of the ion can show the connection level in the corresponding local network. The bigger the ion is, the tighter the connection will be. So, the ion can be used as a measure to detect which community the node belongs to.

Suppose that the neighborhood of node i has the label set $\Omega(i) = (l(i^1), l(i^2), \ldots, l(i^k))$; the new label propagation based on the impact of node is as follows:

$$\text{LP}(i) = \underset{l(i^j) \in \Omega(i)}{\text{argmax}} \left(\text{ion}(i^j)\right). \tag{10}$$

It means that the label of current node is decided by its neighbour which has the biggest impact of node.

We choose the karate network to illustrate the rationality of the impact of node. The karate network was completed by

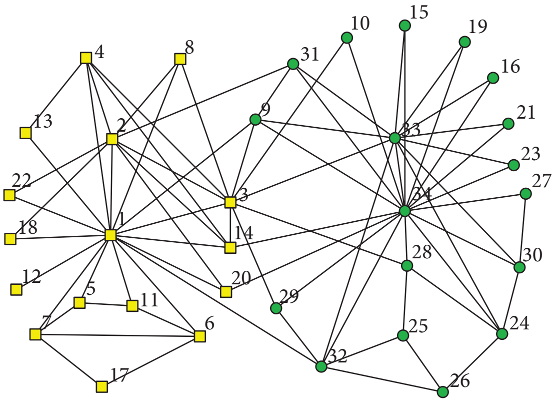


Figure 1: The real community structure of karate network.

Gong et al. after two years' observation. The club splits into two groups because of the dispute between the administrator and the instructor of the club. Figure 1 shows the real community structure of karate network; the nodes with the same mark belong to the same community. Vertices 1 and 34 represent the administrator and the instructor, respectively. When deciding the labels of each node in the community, there are three cases needed to consider. The first case is that when all the neighbors of a node have different labels, then the degree of node will decide the label of the current node; the other case is that when some of the neighbors of the current node share the same label, then the impact of these neighbors on the current node depends on the number of neighbors that share the same label if they have similar degree; otherwise, both of the two factors (degree (i^j) and num $(l(i^j))$) will play important role in deciding the label of the current node. Take node 3 as an example; its neighbors are $\{1, 2, 4, 8, 14, 9, 10, 28, 29, 33\}$, and their degrees are $\{16, 9, 5, 3, 5, 5, 2, 4, 3, 12\}$. If all the labels of these neighbors are different, node 3 will be assigned to the same label with node 1 since the ion of node 1 is the biggest according to our proposed strategy. If all the neighbors of node 3 have been assigned to the right communities, the ion of all these neighbors to node 3 is $\{80, 45, 25, 45, 25, 25, 10, 20, 15, 60\}$; then node 1 has the highest impact on node 3, and node 3 is assigned to the same community with node 1. However, we can not decide the label of node 3 if we use label propagation method in [25] in both cases.

3.3. Particle Swarm Initialization.

3.3. Particle Swarm Initialization. In PSO algorithm, proper initialization scheme can generate a set of high quality solutions and reduce the searching time significantly. Traditional initialization method generates solutions randomly. It does not take the adjacency matrix of the community into consideration, while this matrix usually provides important information to the optimization. In order to make use of this prior knowledge, we put forward a new particle swarm initialization based on the impact of node in this paper. The procedure is as follows.

Step 1. Initialize every node with unique labels, $l(i) = i$.

Step 2. Sort all the nodes in descending order according to their degree.

Step 3. Initialize the ith particle, select the node which has the ith highest degree (denoted as node a), and find the node (denoted as b) which has the maximal degree connected with node a. Assign all the nodes connected with both node a and node b to the same label.

Step 4. Repeat Step 3 until all the networks are gone through or the termination criteria are met.

If the number of nodes is less than particles, select two linked nodes randomly and assign them to the same label for the rest particles. For signed networks, only the positive links are considered in Steps 3 and 4.

The description of proposed initialization method can be seen in Figure 2. According to the adjacency graph, node 6 has the greatest degree among all the nodes, so it is selected firstly. The degree of node 10 is the biggest in all the neighbors of node 6. In this way, nodes (6, 7, 8, 9, 10, 11) will be assigned to the same label. But if we simply assign the nodes connected to node 6 to the same label based on label propagation in [25], the nodes (4, 5, 6, 7, 8, 9, 10, 11) will share the same label. Actually, node 4 and node 5 should have the same label with node 3 according to the structure of the network. The advantage of proposed initialization scheme can be seen from this example.

3.4. Particle Status Updating. Particle status updating is a key process in PSO algorithm. It consists of velocity updating and position updating. In this paper, velocity updating uses the update rule in [25], and the position updating depends on the velocity updating and the proposed impact of node. The velocity updating rule in discrete form is as follows:

$$V_i = \operatorname{sig}\left(\omega V_i + c_1 r_1 \left(pbest \oplus X_i\right) + c_2 r_2 \left(gbest \oplus X_i\right)\right), \quad (11)$$

where ω is the inertia weight and c_1 and c_2 are the cognitive and social components, respectively. r_1 and r_2 are two random numbers which range from 0 to 1. In this paper, the inertia weight ω is randomly generated within $[0, 1]$, and the cognitive and social components c_1 and c_2 are set to the typical value of 1.494 [25]. $\oplus$ is defined as an XOR operator. Suppose that $Y = (y_1, y_2, \ldots, y_n)$ and $X = (x_1, x_2, \ldots, x_n)$; the function $Y = \operatorname{sig}(X)$ is defined as follows:

$$y_i = \begin{cases} 1, & \operatorname{rand}(0,1) < \operatorname{sigmoid}(x_i) \\ 0, & \operatorname{rand}(0,1) \geq \operatorname{sigmoid}(x_i). \end{cases} \quad (12)$$

The sigmoid function is defined as

$$\operatorname{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (13)$$

Now the position updating rule is defined as the following discrete form:

$$X_{i+1} = X_i \otimes V_i, \quad (14)$$

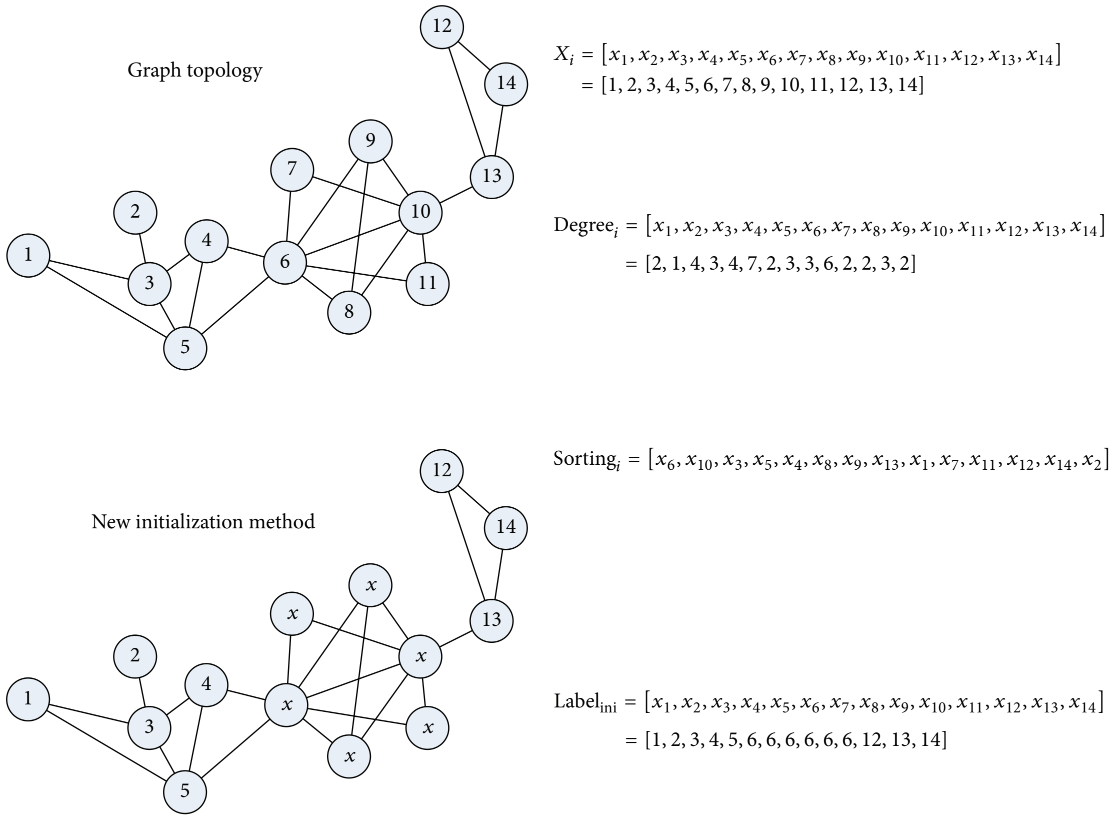

Graph topology
$X_i = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}]$
$= [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]$
$\text{Degree}_i = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}]$
$= [2, 1, 4, 3, 4, 7, 2, 3, 3, 6, 2, 2, 3, 2]$
New initialization method
$\text{Sorting}_i = [x_6, x_{10}, x_3, x_5, x_4, x_8, x_9, x_{13}, x_1, x_7, x_{11}, x_{12}, x_{14}, x_2]$
$\text{Label}_{\text{ini}} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}]$
$= [1, 2, 3, 4, 5, 6, 6, 6, 6, 6, 6, 12, 13, 14]$
1
2
3
4
5
6
7
8
9
10
11
12
13
14
x

Figure 2: Description of proposed initialization method.

where the new position is generated by "position ⊗ velocity," that is, given a position $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and a velocity $V = (v_1, v_2, \ldots, v_n)$. The element of new position $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$ is defined as

$$x_{2i} = \begin{cases} x_{1i}, & v_i = 0 \\ Nbest_i, & v_i = 1, \end{cases} \tag{15}$$

where $Nbest_i$ is calculated by new label propagation in (10) which is based on the impact of node:

$$Nbest_i = \underset{l(i^j)\in\Omega(i)}{\operatorname{argmax}} \left(\text{ion}\left(i^j\right)\right). \tag{16}$$

The procedure of the new label propagation can be described as follows:

(1) Count the labels of all the nodes connected with the current node, which are called neighborhood nodes in the following steps.

(2) Calculate the impact of these nodes according to (9).

(3) Find the label of node which has the greatest ion and assign this label to the current node; if there are more than 2 labels which have the same votes, select one label randomly to assign it to the current node.

It is noted that this update strategy is more effective in the later stage of community detection because we often encounter the situation that a node's neighbors belong to different communities. However, we do not know when the algorithm converges to a stable state on different networks. So, we use the update strategy in [25] in uneven number generation because of its simplicity, and we use the proposed update strategy in even number generation because of its effectivity.

3.5. Framework of the Proposed Algorithm. After illustrating all the corresponding strategies in detail, the framework of the proposed algorithm is presented in Algorithm 1.

3.6. Complexity Analysis. The main complexity of the proposed algorithm relies on the cycling process. Suppose that the network we tested has n nodes and m edges. In Algorithm 1, Step 1 can be finished in linear time. The time complexity of Steps 2.1 and 2.2 is $O(n)$ and $O(2m)$ in the worst case, respectively. Step 2.3 can be accomplished in $O(m + n)$ time; Steps 2.4 and 2.5 need $O(1)$ operation. So, the total time complexity of the proposed algorithm is $O(\text{pop} \times \text{maxgen} \times \max\{(m+n), (2m), O(n)\})$, which can be simplified as $O(\text{pop} \times \text{maxgen} \times (2m))$.

4. Experiment Discussion

In this section, detailed experiments are carried out to test the effectiveness of the proposed algorithm against other five algorithms or methods, including CNM, FTQ, Informap,

```
(1) Input:
    Adjacency matrix $A$;
    Population size: pop;
    Maximum generation: maxgen;
    Tuning parameter: $\lambda$;
(2) Step 1: Initialization
    (1.1) Position initialization: $\mathbf{x} = (x_1, x_2, \ldots, x_{\text{pop}})^T$;
    (1.2) Velocity initialization: $\mathbf{v} = (v_1, v_2, \ldots, v_{\text{pop}})^T$;
    (1.3) Personal best position initialization: $\mathbf{pbest} = (pbest_1, pbest_2, \ldots, pbest_{\text{pop}})^T$, $\mathbf{pbest} = \mathbf{x}$;
    (1.4) Global best position initialization: select the global best position gbest in pbest, $\mathbf{gbest} = (gbest_1, gbest_2, \ldots, gbest_{\text{pop}})^T$.
(3) Step 2: Iteration
    (2.1) Calculate new velocity according to (11), $\mathbf{v}^{t+1} = (v_1^{t+1}, v_2^{t+1}, \ldots, v_{\text{pop}}^{t+1})^T$;
    (2.2) Calculate new position according to (14), (15), and (16)
          $\mathbf{x}^{t+1} = (x_1^{t+1}, x_2^{t+1}, \ldots, x_{\text{pop}}^{t+1})^T$;
          if $\text{mod}(t, 2) = 0$
             Proposed strategy is adopted
          else
             Strategy in [25] is adopted
          endif
    (2.3) Function evaluation;
    (2.4) Update personal best position pbest;
    (2.5) Update global best position gbest.
(4) Step 3: Termination Criteria
    if maxgen is arrived
      Stop and output gbest;
    else
      Go back to Step 2;
    end if
(5) Output: gbest.
```

Algorithm 1: Framework of the proposed algorithm.

and other two nature inspired algorithms GA-net and label propagation in Gong et al.'s literature [25].

CNM was presented by Clauset et al. in [10]. This method is essentially a fast implementation of the GN approach. GA-net is a single-objective algorithm proposed by Pizzuti in [14]. The algorithm optimizes a simple but efficacious fitness function, which is called community score, to identify densely connected groups. Fine-tuned modularity density algorithm (denoted as FTQ) is a fine-tuned algorithm based on modularity density proposed by Chen et al.; it detects the community structure via splitting and merging the network [6]. Informap is introduced by Rosvall and Bergstrom in [27]. This algorithm uses a new information theoretic approach to reveal community structure in weighted and directed networks. In [25], a multiobjective algorithm based on MOEA/D and PSO is combined to detect the structure of community. Considering the objective function we adopted is a single-objective function, only the label propagation and particles updating strategy in [25] are used and combined with modularity density function (denoted as DPSO in the following) to compare with proposed algorithm.

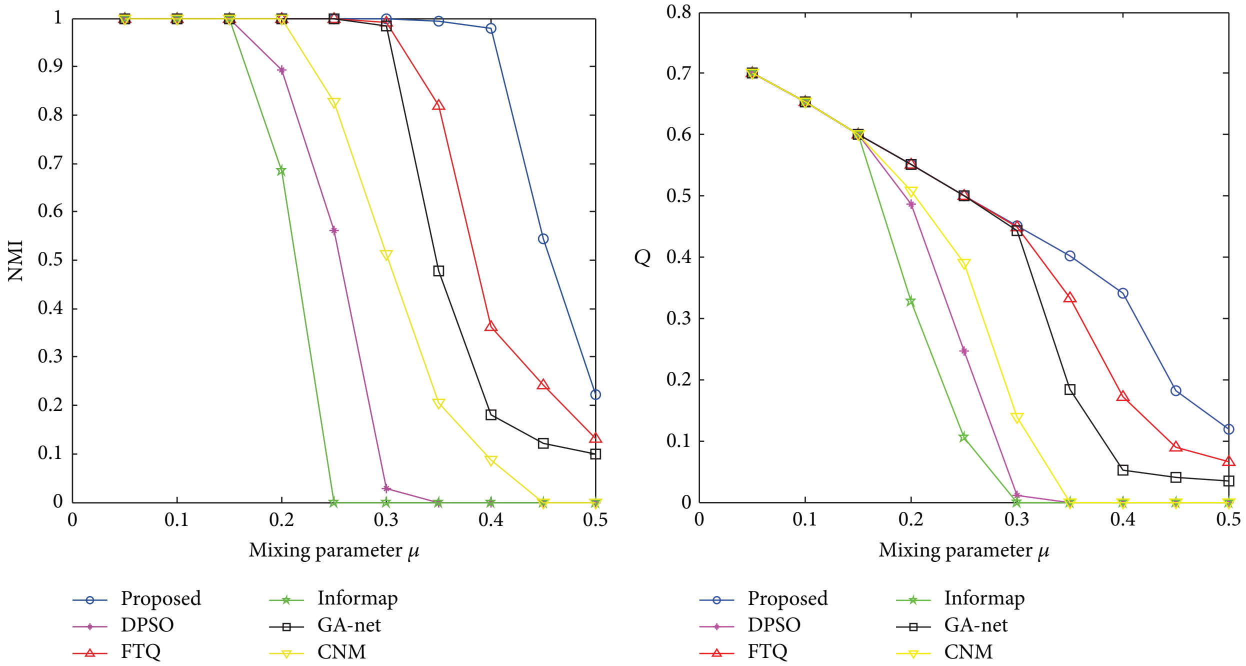
4.1. Experimental Settings. In this paper, we choose *modularity Q* and *Normalized Mutual Information* (NMI) as the measurement when the ground truth of a network is known. Otherwise, only modularity Q is adopted. For signed networks, SQ is adopted to evaluate the performance in [28], which is formulized as

$$\text{SQ} = \frac{1}{2\omega^+ + 2\omega^-} \sum_{i,j} \left(\omega_{ij} - \left(\frac{\omega_i^+ \omega_j^+}{2\omega^+} - \frac{\omega_i^- \omega_j^-}{2\omega^-} \right) \right) \delta(i, j), \tag{17}$$

where ω_i^+ and ω_i^- represent the sum of all positive and negative weights of node i, respectively, and ω_{ij} represents the weight of adjacency matrix of the signed network.

The *Normalized Mutual Information* (NMI) described in [29] is an index to estimate the similarity between two community detection results. If we assume that A, B are the two partitions of a network and C is the confusion matrix, then NMI $\in [0, 1]$ measures the similarity between A and B. The larger the value of NMI is, the more similar the partitions A and B are. If $A = B$, $\text{NMI}(A, B) = 1$; if A and B are completely different, $\text{NMI}(A, B) = 0$. $\text{NMI}(A, B)$ can be calculated as follows:

$$\text{NMI}(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log\left(C_{ij} N / C_{i.} C_{.j}\right)}{\sum_{i=1}^{C_A} C_{i.} \log\left(C_{i.}/N\right) + \sum_{j=1}^{C_B} C_{.j} \log\left(C_{.j}/N\right)}, \tag{18}$$



(a)

(b)

Figure 3: Average maximum NMI and Q values over 30 runs on GN extended benchmark networks.

where C_A (C_B) is the number of the communities in partition A (B) and $C_{i.}$ ($C_{.j}$) is the sum of the elements of C in row i (column j).

In this paper, parameter λ in objective function increases from 0.3 to 0.8 with interval 0.1, and parameter γ in GA-net ranges from 1 to 1.5 with interval 0.1. For each value of parameter λ or γ, all the algorithms run 30 independent times on the test problems with setting pop to 100 and maxgen to 100. Among all the results for each network, the best one is selected and shown in the following experiments.

4.2. Experiments on GN Extended Benchmark Networks. GN extended benchmark network was proposed by Lancichinetti et al. in [30], which is an extension of the classical GN benchmark proposed by Girvan and Newman in [4]. GN extended benchmark network consists of 128 nodes divided into four communities, and each community has 32 nodes. The average degree of node is 16, and mixing parameter μ decides the percentage of connections between communities to the total connections. When $\mu < 0.5$, the network has strong community structure. On the contrary, when $\mu > 0.5$, the community structure is vague, and it is hard to detect its structure.

Experiments on GN extended benchmark networks are done to test the performance of our algorithm. CNM, FTQ, GA-net, Informap, DPSO, and the proposed algorithms are tested on 10 GN extended networks with mixing parameter μ ranging from 0.05 to 0.5. Figure 3 shows the average maximum NMI and Q values obtained from different algorithms when the mixing parameter μ increases from 0.05 to 0.5 with interval 0.05.

As is shown in Figure 3(a), when the mixing parameter $\mu \leq 0.15$, Informap, FTQ, CNM, GA-net, DPSO, and the proposed algorithm can figure out the true partition (NMI equals 1). With the mixing parameter increasing, the community structure of the network becomes fuzzy and it becomes difficult to detect the true structure of the community. Informap and DPSO first show their weakness, and their detection ability decreases rapidly from $\mu = 0.15$ to 0.3. Then detection ability of CNM decreases from $\mu = 0.2$. When $\mu > 0.3$, GA-net and FTQ show its limitation in detecting the community structure. Compared with them, the proposed algorithm shows its superiority. As seen from Figure 3(b), the same conclusion could be derived from measurement Q.

More experiments are discussed on GN extended benchmark network in detail to illustrate the performance of proposed algorithm. In our objective function, λ is a tuning parameter. The bigger the λ is, the more the number of the communities will be detected generally.

Figure 4 shows how NMI changes with mixing parameter when λ adopts different values. As seen from them, DPSO almost gets the same NMI values when λ ranges from 0.3 to 0.8. On the contrary, the proposed algorithm shows its excellent detection ability when community structure gets more and more obscure with the changing of parameter λ. From these figures, the superiority of proposed algorithm is demonstrated. In our view, the designed definition takes the topology of community structure into consideration, which allows the algorithm to detect more obscure structure than the others with a suitable tuning parameter value in the objective functions.

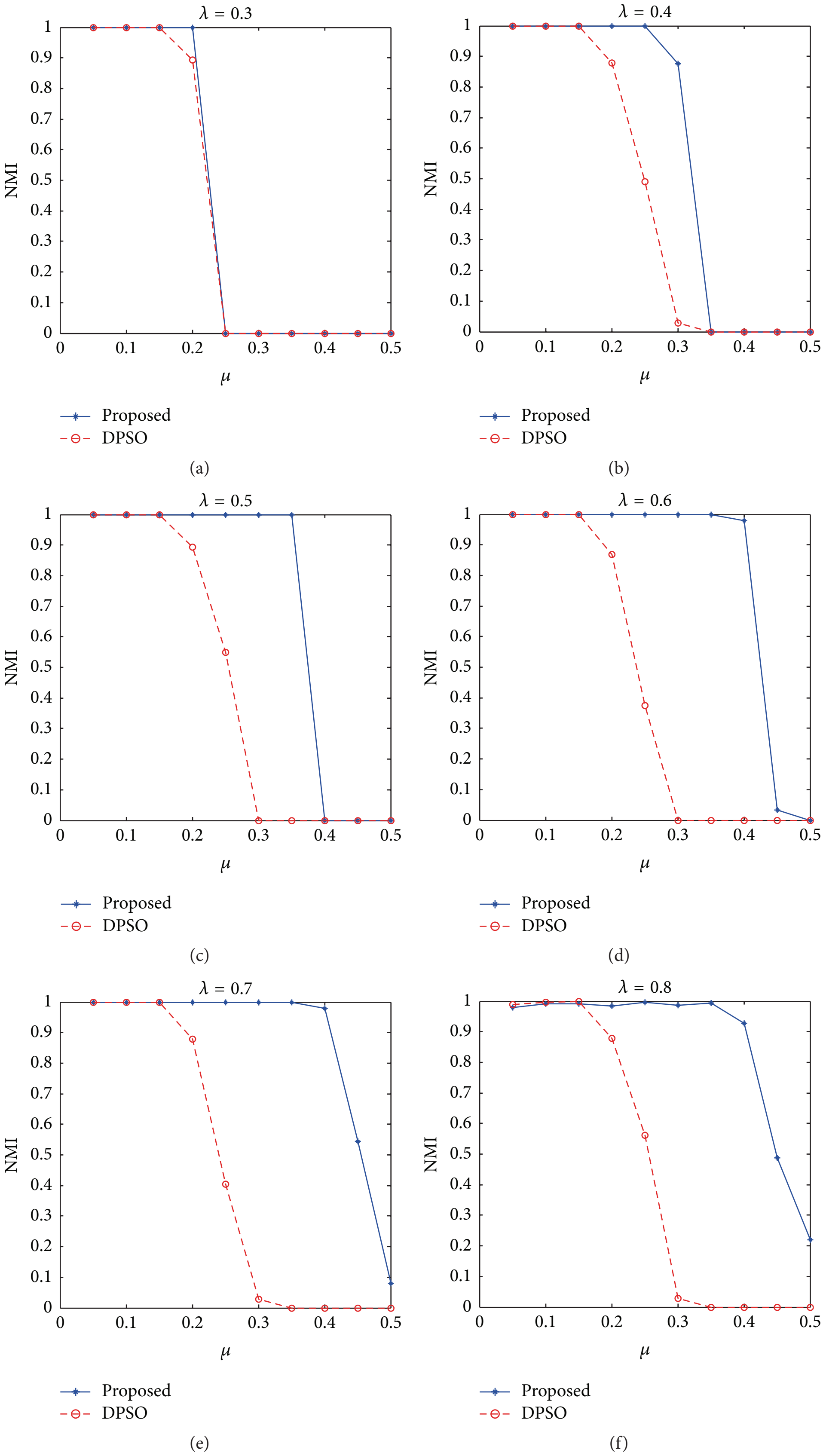
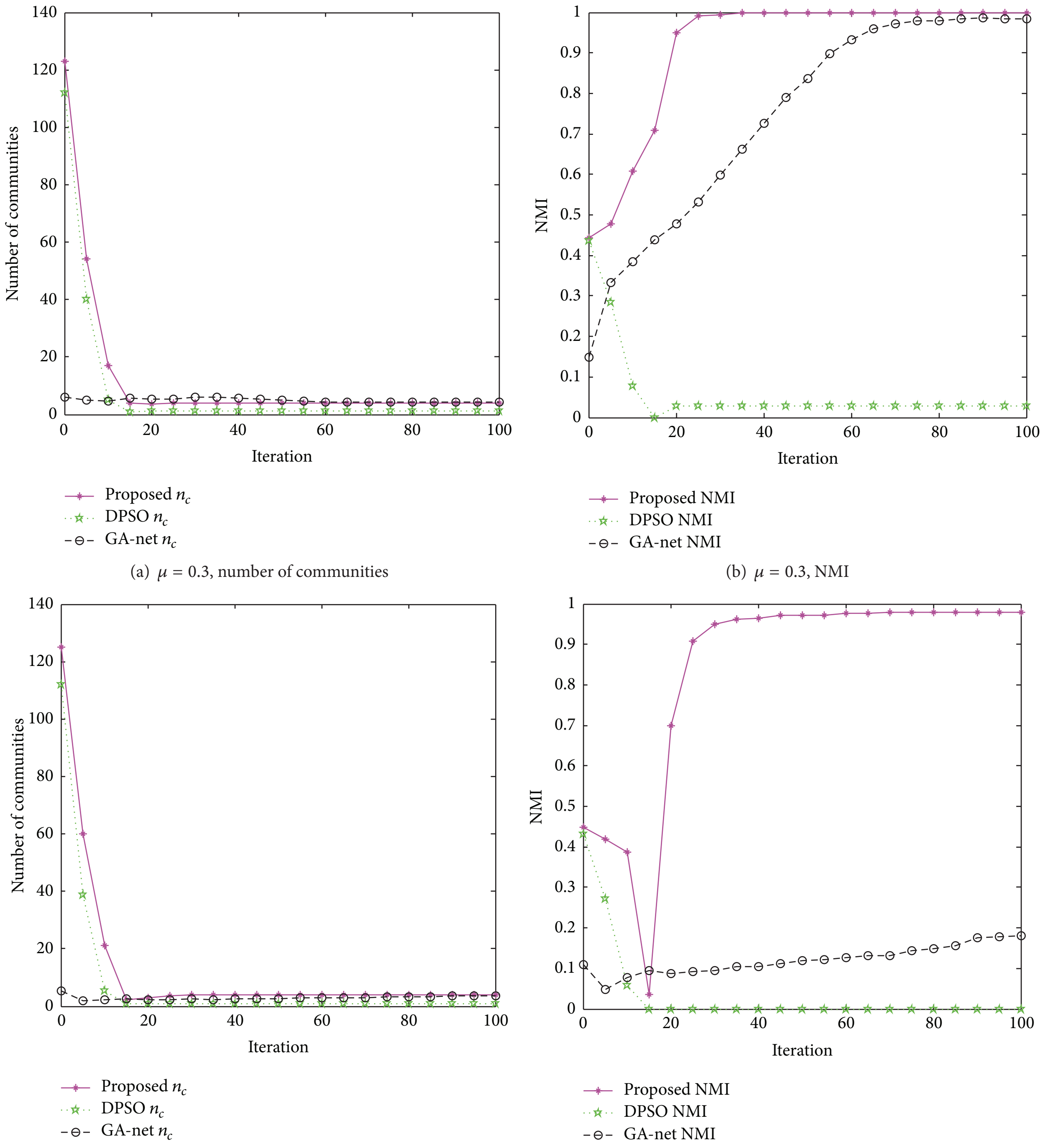


Figure 4: Average maximum NMI values over 30 runs on GN extended benchmark networks with different λ values.



(a) $\mu = 0.3$, number of communities

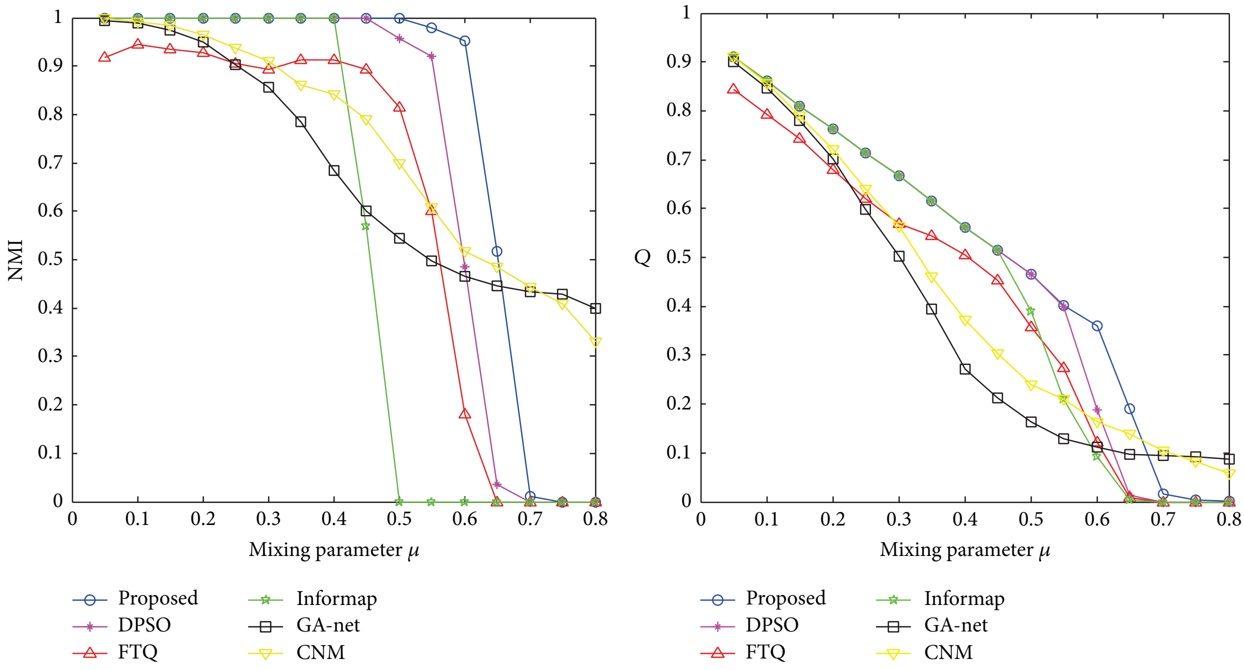
(b) $\mu = 0.3$, NMI

(c) $\mu = 0.4$, number of communities

(d) $\mu = 0.4$, NMI

Figure 5: Convergence of three algorithms.

In order to discuss the convergence of the nature inspired algorithms (the proposed algorithm, DPSO, and GA-net algorithm), we choose GN extended benchmark networks $\mu = 0.3$ and $\mu = 0.4$ to illustrate it. As shown in Figures 5(a) and 5(c), the number of communities (denoted as n_c in the figures) obtained from all three algorithms converges to a stable value no matter $\mu = 0.3$ or 0.4. During the optimization process, the number of communities has converged to a stable value since the 20th iteration. At the same time, the labels of nodes still keep changing to obtain better results. At last, NMI converges to 1 or almost 1 in about the 30th iteration in Figures 5(b) and 5(d). For DPSO, NMI values are quite low (about equal to 0) since it can not detect the exact number of communities. For GA-net, when $\mu = 0.3$, it can obtain a satisfying result, but it has a slow convergence speed. When the mixing parameter increases to 0.4, it fails



(a)

(b)

FIGURE 6: Average maximum NMI and Q values over 30 runs on LFR benchmark networks.

to detect community structure exactly. From Figure 5, a conclusion can be derived that the proposed algorithm significantly outperforms other algorithms and can detect the real structure of a network effectively.

4.3. Experiments on LFR Benchmark Network. On the GN extended benchmark network, all communities have exactly the same size, and the size of network is small, so it cannot reflect some important features in real networks. Because of this, LFR benchmark networks are proposed by Lancichinetti et al. in [30], in which the parameters α and β are set to tune the distribution of degree and community size. Each node shares a fraction $1 - \mu$ of its links with the other nodes in the same community and a fraction μ with the nodes in other communities.

In this paper, we use 16 LFR benchmark networks; their mixing parameter increases from 0.05 to 0.8 with interval 0.05. Each of them consists of 1000 nodes and the cluster size ranges from 10 to 50, $\alpha = 2$ and $\beta = 1$. The averaged degree for each node is 20 and the maximum node degree is 50.

TABLE 1: Characteristics of adopted real-life networks.

Network	Nodes	Edges	Real communities
Karate	34	78	2
Dolphin	62	159	2
Football	115	616	12
Politics	105	613	3
SFI	118	200	Unknown
Netscience	1589	2742	Unknown

As is shown in Figure 6, the proposed algorithm and DPSO have a better performance than the other 4 algorithms on LFR networks because the proposed algorithm and DPSO share the same objective function. What is more, the detection ability of the proposed algorithm is stronger than that of DPSO with the mixing parameter μ increasing. A conclusion that our initialization and updating schemes can reveal the structure of community more exactly than label propagation in DPSO can be obtained from the comparison result. The performance of GA-net is the worst in all the 6 algorithms when $\mu \leq 0.65$. With the increasing of mixing parameter, GA-net has the best performance when $\mu \geq 0.65$, for the reason that it uses different encoding scheme. DPSO and our proposed algorithm tend to consider all the nodes as a whole community since the structure is too obscure. The experiment on LFR networks also shows the excellent detection ability of the proposed algorithm.

4.4. Experiments on Real-Life Networks. We apply our algorithm to six well-known real-life networks, including Zacharys Karate Club network [31], Dolphin social network [32], American College Football network, Krebs Books on US Politics network, Santa Fe Institute (SFI) network [4], and Netscience network [33]. The characteristics of the networks are shown in Table 1.

Comparison experiments on six real-life networks are tested and shown in Table 2. From what is recorded on the Zacharys Karate Club, Dolphin, and American College Football networks, the proposed algorithm shows a better performance than the other five algorithms. Referring to the Krebs' Books on US Politics network, the performance of the

Table 2: Experiment result on real-life networks.

Dataset	Measurement	Proposed	DPSO	GA-net	FTQ	CNM	Informap
Karate	$\mathrm{NMI}_{\max}$	**1.000**	1.000	0.6369	0.7467	0.7154	0.7198
	$\mathrm{NMI}_{\mathrm{avg}}$	**1.000**	0.7369	0.6369	0.6956	0.7013	0.7142
	$Q_{\max}$	0.4037	**0.4329**	0.4060	0.3913	0.4095	0.4291
	Q_{avg}	0.3835	**0.4132**	0.4060	0.3796	0.3916	0.4031
Dolphin	$\mathrm{NMI}_{\max}$	**1.000**	1.000	0.4236	0.8794	0.6049	0.6395
	$\mathrm{NMI}_{\mathrm{avg}}$	**0.9610**	0.9477	0.4060	0.8576	0.5893	0.6043
	$Q_{\max}$	**0.5178**	0.5136	0.4634	0.4914	0.4018	0.4839
	Q_{avg}	**0.5114**	0.5109	0.4634	0.4837	0.3945	0.4839
Politics	$\mathrm{NMI}_{\max}$	0.5916	0.5745	0.4403	**0.6792**	0.5701	0.6216
	$\mathrm{NMI}_{\mathrm{avg}}$	0.5763	0.5745	0.4201	**0.6513**	0.5620	0.6165
	$Q_{\max}$	0.5036	0.5256	0.5035	**0.5805**	0.5513	0.5674
	Q_{avg}	0.4969	0.5235	0.4798	**0.5604**	0.5470	0.5364
Football	$\mathrm{NMI}_{\max}$	**0.9253**	0.9252	0.9252	0.9013	0.7438	0.6574
	$\mathrm{NMI}_{\mathrm{avg}}$	**0.9141**	0.9094	0.9107	0.8984	0.7438	0.6574
	$Q_{\max}$	**0.6093**	0.6037	0.6012	0.4892	0.5394	0.3484
	Q_{avg}	**0.6021**	0.6015	0.5906	0.4519	0.5394	0.3484
SFI	$Q_{\max}$	0.7325	**0.7489**	0.5716	0.6946	0.4865	0.5319
	Q_{avg}	**0.7287**	0.7252	0.5620	0.6743	0.4617	0.5319
Netscience	$Q_{\max}$	**0.9474**	0.9206	0.8739	0.9031	0.8021	0.7183
	Q_{avg}	**0.9400**	0.9174	0.8591	0.8917	0.7834	0.7012

proposed algorithm is little worse than FTQ and Informap algorithm. For the two real-life datasets without ground truth labels, SFI network and the Netscience network, the proposed algorithm outperforms the other algorithms, too.

In order to give a time cost analysis of the proposed algorithm, an experiment on the time comparison of three nature inspired algorithms is carried out in Figure 7. We can see that DPSO and the proposed algorithm have comparative time cost, but GA-net is the most time-consuming since the time complexity of decoding step is more than that of the others.

A further analysis on the effect of the proposed definition (impact of the node) and the adopted objective function is carried out on real-life networks. Figure 8 shows the experiment result obtained from different initialization and updating strategies; (a)~(d) represent the results of the proposed initialization and the proposed updating strategy, the proposed initialization and the updating strategy in [25], the initialization strategy in [25] and the proposed updating strategy, the initialization in [25] and the updating strategy in [25], respectively. When comparing (a) to (b) and (c) to (d), we can clearly see that the updating strategy has a better performance than the other strategy. The designed initialization strategy shows a weak superiority effect compared to other initialization strategies when comparing (a) to (c) and (b) to (d), which indicates that the proposed initialization strategy can generate a set of solutions with better quality than others. From this figure, we can also derive that the proposed algorithm have a faster convergence speed than the other strategies.

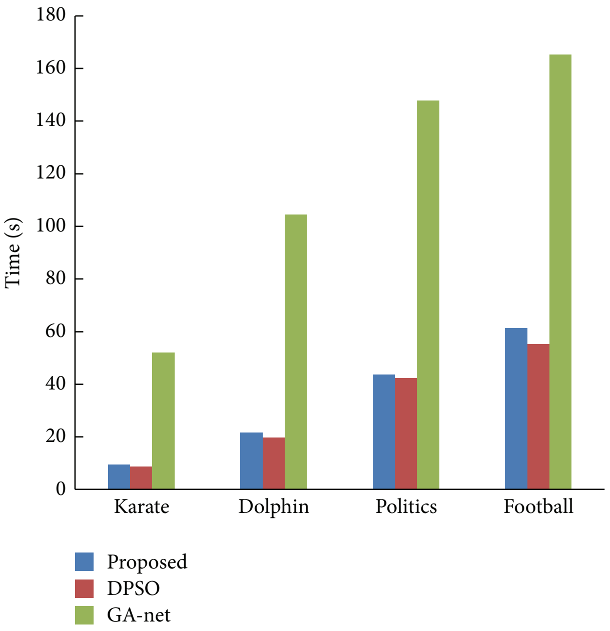
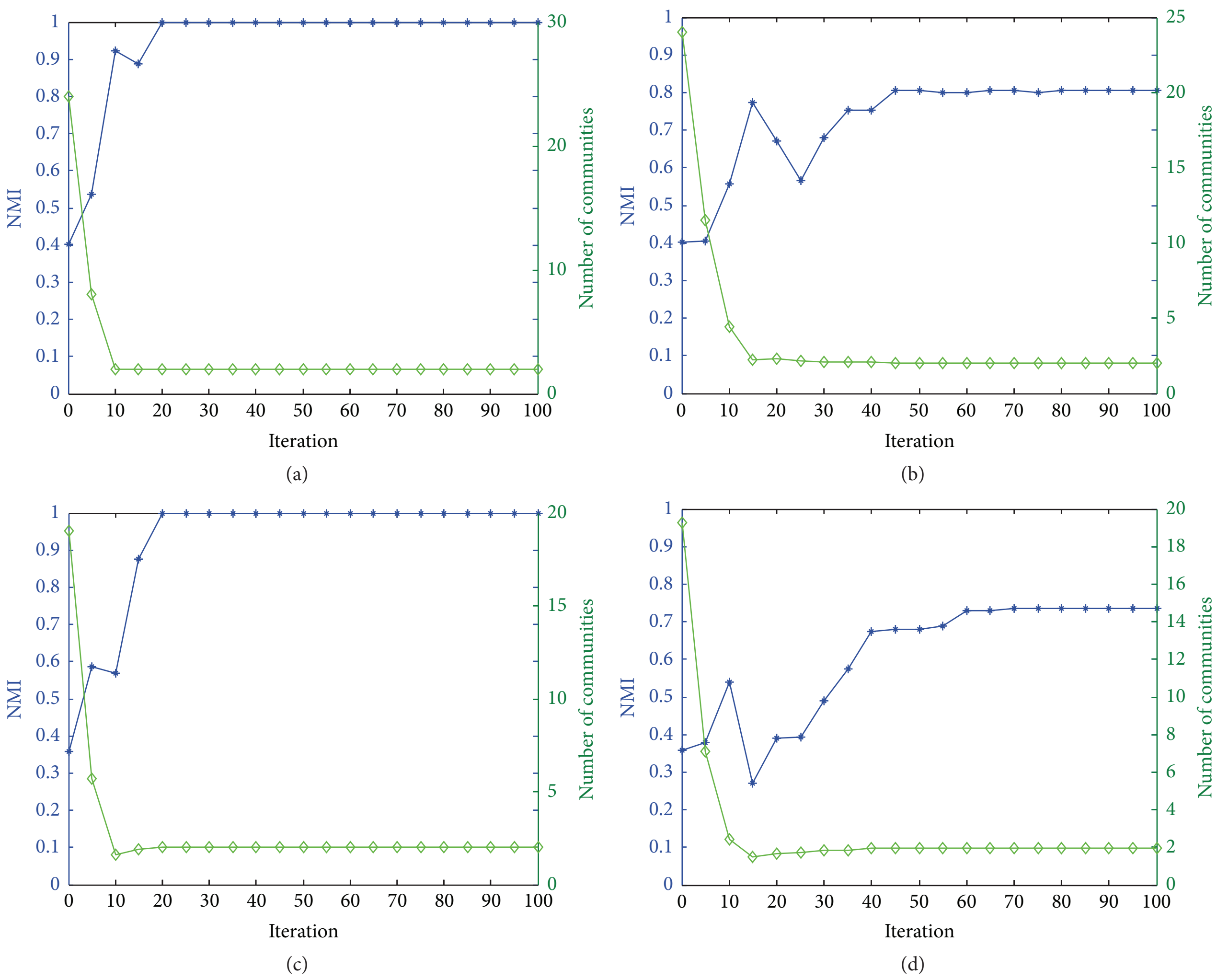


Figure 7: Computational time of three algorithms.

In order to analyse the benefit of the adopted modularity density objective function, we give a comparison experiment obtained from two different objective functions: modularity density and community score (GA-net adopted) in Figure 9. It shows that when we use the designed initialization and updating strategy, modularity density function is more effective than community score function.



(a)

(b)

(c)

(d)

Figure 8: The comparison of different initialization and update strategies (the curve marked in green is the average number of communities and that in blue is the average NMI values during the optimization process).

Figure 10 displays a visible detection result obtained from the proposed algorithm on Zacharys Karate Club network. As is shown in this figure, the proposed algorithm figures out the true partition of the network when $\lambda = 0.3$. In the partition with $\lambda = 0.8$, our algorithm detects 4 communities, which is still reasonable since the original one community is divided into two subcommunities.

Table 3: Experiment results on signed networks.

Dataset	Proposed		DPSO	
	NMI	SQ	NMI	SQ
SPP	**1**	**0.4547**	1	0.4547
GGS	**1**	**0.4310**	1	0.4310
Illustrate	**1**	**0.5643**	0.8900	0.5406

4.5. Experiments on Signed Networks. In this subsection, we apply the proposed algorithm to three real-world signed networks, including the illustrative signed network [34], the Slovene Parliamentary Party (SPP) network [35], and the Gahuku-Gama Subtribes (GGS) network [36].

Only DPSO and the proposed algorithm are considered on the three real-world signed networks because of the limitation of objective functions in the other algorithms. The best average statistical result is selected to be shown in Table 3.

As is shown in Table 3, we can clearly notice that both DPSO and the proposed algorithm can successfully detect the community structure of the network (NMI = 1) on SPP and GGS, but for the Illustrate network, the proposed algorithm shows a better performance than DPSO.

SPP consists of 10 Slovene Parliamentary Parties set up by a series of experts on parliamentary activities in 1994. The topology community structure of the SPP network recognized by our algorithm is shown in Figure 11, in which the network is divided into two subcommunities when $\lambda = 0.3$.

GGS consists of 16 nodes which represent 16 Gahuku-Gama Subtribes involved in the warfare distributed in a particular area. The detection results of the proposed

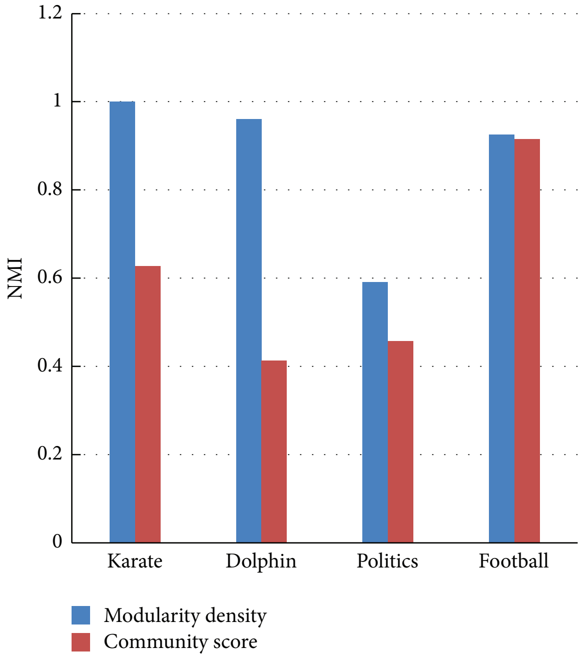
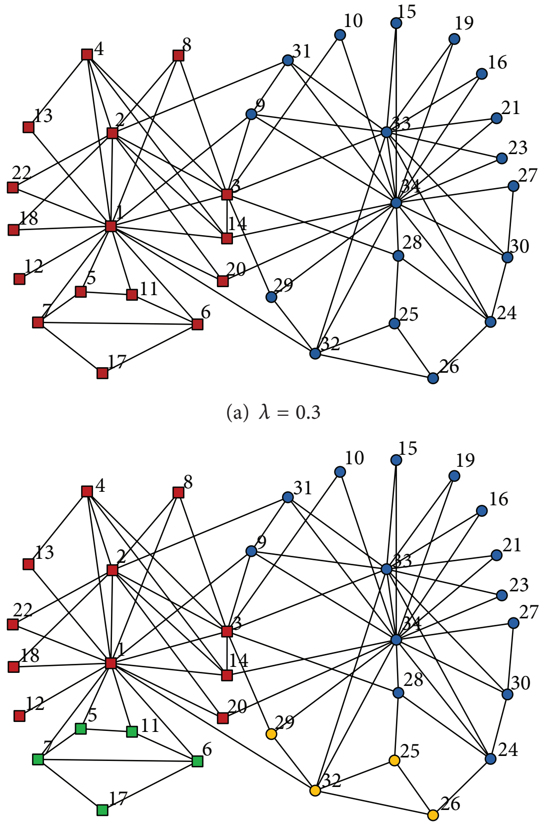


Figure 9: Benefit of modularity density.



(a) $\lambda = 0.3$

(b) $\lambda = 0.8$

Figure 10: Community structure detected by the proposed algorithm on Zacharys Karate Club.

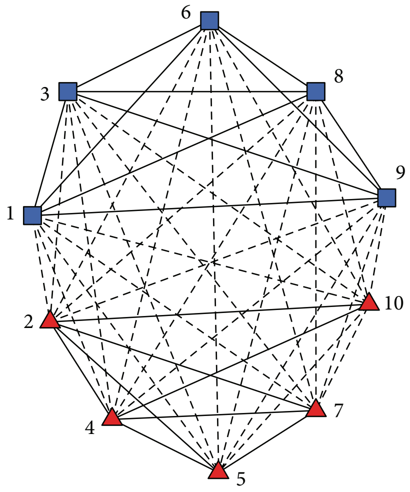


Figure 11: Community structure detected by proposed algorithm on SPP.

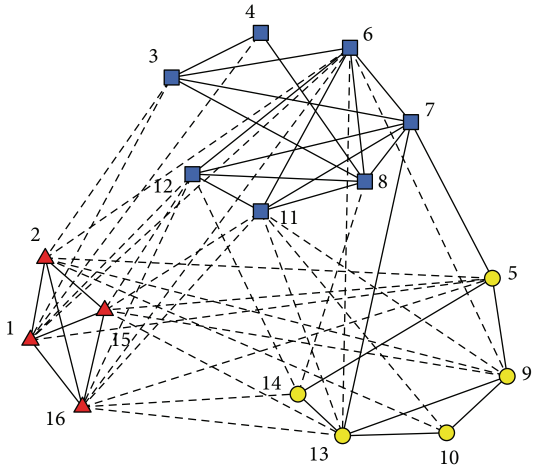


Figure 12: Community structure detected by the proposed algorithm on GGS.

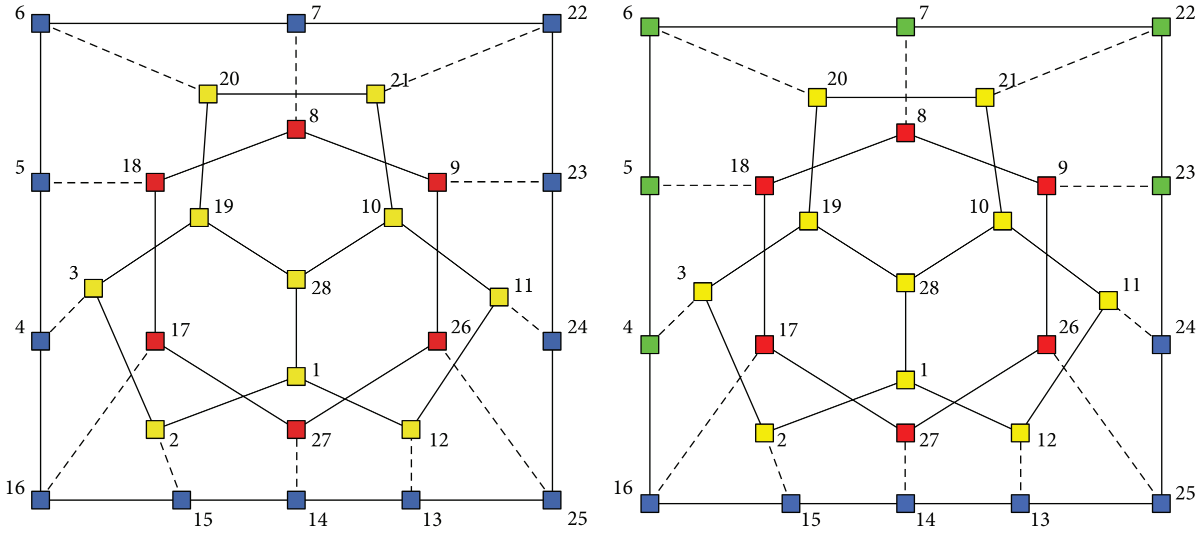
algorithm are displayed in Figure 12. Our algorithm detects three communities when $\lambda = 0.3$.

The illustrative signed network consists of 28 nodes and is divided into three communities. The community structure of it detected by the proposed algorithm is different when the value of parameter λ differs, in which the network is divided into three subcommunities with $\lambda = 0.2$ and four subcommunities with $\lambda = 0.3$. The corresponding result is shown in Figure 13.

From all the above analysis, the results on synthetic and real-life networks show that the proposed algorithm can deal with community detection problems effectively and promisingly.

5. Conclusion

After showing detailed description and experiments analysis in this paper, a conclusion about it can be drawn.



(a) $\lambda = 0.2$

(b) $\lambda = 0.3$

Figure 13: Community structure detected by the proposed algorithm on Illustrate.

Firstly, the definition of the impact of node is proposed and the new label propagation based on it is designed. This new definition demonstrates its effectiveness by experiments on synthetic and real-life networks.

Secondly, special initialization and updating strategies based on the impact of node are designed. By using these strategies, the structure of the community is detected more exactly compared with other methods.

Thirdly, we modify modularity density function to signed networks according to the character of networks. Combining proposed initialization and updating strategies with this objective function, it can detect the community structure of signed networks exactly.

Additionally, the proposed strategy can be implemented into other graph-related fields easily. This paper adopts single-objective function, modularity density, to detect community structure, which can be easily extended to a multiobjective optimization problem. We believe that it would show its superiority to others in this way. Moreover, applying community detection to our radar communication networks to improve the cooperation quality is also one of our future works.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, “Natural communities in large linked networks,” in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 541–546, ACM, Washington, DC, USA, August 2003.

[2] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On power-law relationships of the Internet topology,” in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM ’99)*, pp. 251–262, ACM, Cambridge, Mass, USA, August-September 1999.

[3] S. Lozano, J. Duch, and A. Arenas, “Analysis of large social datasets by community detection,” *The European Physical Journal Special Topics*, vol. 143, no. 1, pp. 257–259, 2007.

[4] M. Girvan and M. E. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[5] C. A. Hidalgo, B. Winger, A.-L. Barabási, and R. Hausmann, “The product space conditions the development of nations,” *Science*, vol. 317, no. 5837, pp. 482–487, 2007.

[6] M. Chen, K. Kuzmin, and B. K. Szymanski, “Community detection via maximization of modularity and its variants,” *IEEE Transactions on Computational Social Systems*, vol. 1, no. 1, pp. 46–65, 2014.

[7] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[8] U. Elsner, “Graph partitioning—a survey,” in *Encyclopedia of Parallel Computing*, pp. 805–808, Springer US, 1997.

[9] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, Article ID 066133, 5 pages, 2004.

[10] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Physical Review E*, vol. 70, no. 6, Article ID 066111, 2004.

[11] R. Guimerà and L. A. N. Amaral, “Functional cartography of complex metabolic networks,” *Nature*, vol. 433, no. 7028, pp. 895–900, 2005.

[12] J. Duch and A. Arenas, “Community detection in complex networks using extremal optimization,” *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 72, no. 2, Article ID 027104, 2005.

[13] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.

[14] C. Pizzuti, "GA-Net: a genetic algorithm for community detection in social networks," in *Parallel Problem Solving from Nature—PPSN X*, pp. 1081–1090, Springer, Berlin, Germany, 2008.

[15] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 418–430, 2012.

[16] A. Arenas and A. Díaz-Guilera, "Synchronization and modularity in complex networks," *The European Physical Journal: Special Topics*, vol. 143, no. 1, pp. 19–25, 2007.

[17] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 36–41, 2007.

[18] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, "Quantitative function for community detection," *Physical Review E*, vol. 77, no. 3, Article ID 036109, 2008.

[19] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '95)*, vol. 1, pp. 39–43, Nagoya, Japan, October 1995.

[20] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[21] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "A fast particle swarm algorithm for solving smooth and non-smooth economic dispatch problems," *Engineering Optimization*, vol. 43, no. 5, pp. 485–505, 2011.

[22] Z. Zhu, J. Zhou, Z. Ji, and Y.-H. Shi, "DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 643–658, 2011.

[23] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, vol. 5, pp. 4104–4108, Orlando, Fla, USA, October 1997.

[24] W.-N. Chen, J. Zhang, H. S. H. Chung, W.-L. Zhong, W.-G. Wu, and Y.-H. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 2, pp. 278–300, 2010.

[25] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 82–97, 2013.

[26] D. Aloise, G. Caporossi, P. Hansen, L. Liberti, S. Perron, and M. Ruiz, "Modularity maximization in networks by variable neighborhood search," in *Proceedings of the 10th DIMACS Implementation Challenge Graph Partitioning and Graph Clustering*, pp. 113–127, Atlanta, Ga, USA, February 2012.

[27] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–1123, 2007.

[28] S. Gómez, P. Jensen, and A. Arenas, "Analysis of community structure in networks of correlated data," *Physical Review E*, vol. 80, no. 1, Article ID 016114, 2009.

[29] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach," *The European Physical Journal B—Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 331–338, 2004.

[30] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, Article ID 046110, 2008.

[31] W. Zachary, "An information flow modelfor conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.

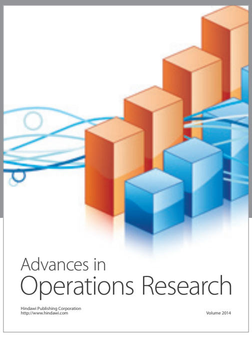
[32] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

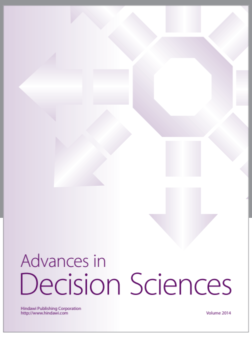
[33] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, Article ID 036104, 2006.

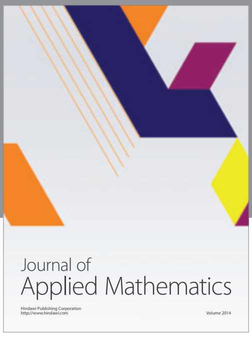
[34] B. Yang, W. K. Cheung, and J. Liu, "Community mining from signed social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 10, pp. 1333–1348, 2007.

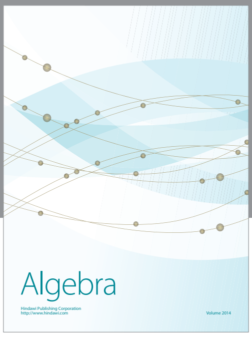
[35] A. Ferligoj and A. Kramberger, *An Analysis of the Slovene Parliamentary Parties Network*, 1996.

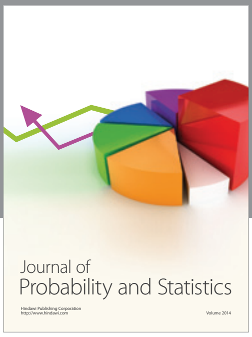
[36] K. E. Read, "Cultures of the central highlands, new Guinea," *Southwestern Journal of Anthropology*, vol. 10, no. 1, pp. 1–43, 1954.


Advances in Operations Research

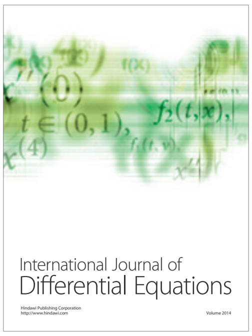

Advances in Decision Sciences


Journal of Applied Mathematics

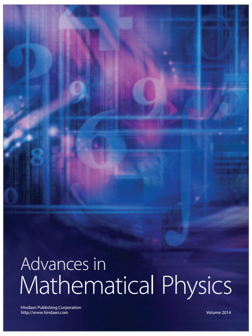

Algebra

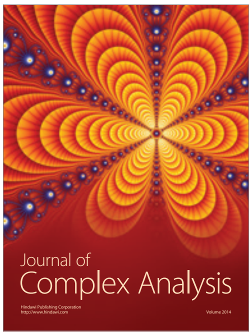

Journal of Probability and Statistics

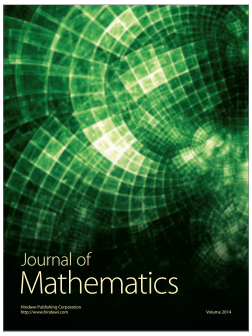
The Scientific World Journal

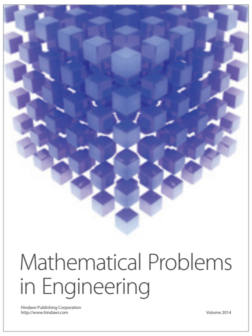

International Journal of Differential Equations

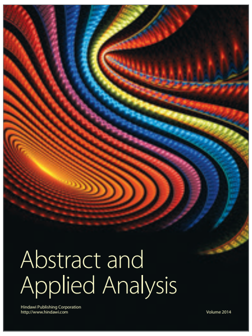
International Journal of Combinatorics

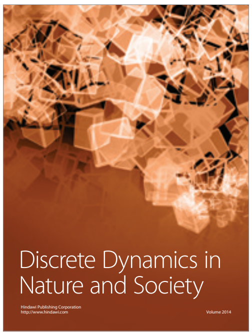

Advances in Mathematical Physics

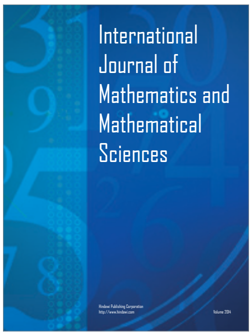

Journal of Complex Analysis


Journal of Mathematics


Mathematical Problems in Engineering

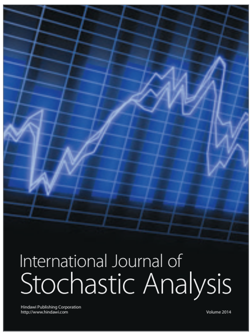

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


International Journal of Stochastic Analysis

Journal of Optimization