Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2015, Article ID 273054, 12 pages http://dx.doi.org/10.1155/2015/273054



Research Article

Hybrid Self-Adaptive Algorithm for Community Detection in Complex Networks

Bin Xu,^{1,2} Jin Qi,^{1,2} Chunxia Zhou,¹ Xiaoxuan Hu,¹ Bianjia Xu,¹ and Yanfei Sun³

¹Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing 210003, China ²School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China ³School of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Correspondence should be addressed to Yanfei Sun; sunyanfei@njupt.edu.cn

Received 19 June 2015; Accepted 17 August 2015

Academic Editor: William Guo

Copyright © 2015 Bin Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The study of community detection algorithms in complex networks has been very active in the past several years. In this paper, a Hybrid Self-adaptive Community Detection Algorithm (HSCDA) based on modularity is put forward first. In HSCDA, three different crossover and two different mutation operators for community detection are designed and then combined to form a strategy pool, in which the strategies will be selected probabilistically based on statistical self-adaptive learning framework. Then, by adopting the best evolving strategy in HSCDA, a Multiobjective Community Detection Algorithm (MCDA) based on kernel *k*-means (KKM) and ratio cut (RC) objective functions is proposed which efficiently make use of recommendation of strategy by statistical self-adaptive learning framework, thus assisting the process of community detection. Experimental results on artificial and real networks show that the proposed algorithms achieve a better performance compared with similar state-of-the-art approaches.

1. Introduction

Since many complex systems, such as the Internet, social networks, and biological networks, can be modeled as complex networks, the study of complex networks is essential to better understand and analyze such systems. In complex networks, community structure [1] refers to the node groups which have the feature that connections between the nodes in the same group are dense and connections between different groups are sparse. In addition to the properties of small world, scale-free, and high clustering coefficient, community structure is another important feature of complex networks. Community detection [2] (also known as network clustering, graph clustering) is to find a division of nodes to obtain community structure. Community detection is helpful to better understand the topology and functions of complex networks [3]. For example, mining community structure on the Internet can not only improve the web search results and enhance the user experience but also implement the hot

topic tracking system. Obtaining the community structure of social networks can help to find social circles with the same hobbies. Therefore, it is essential to further study the community detection in complex networks.

Fortunato [2], Schaeffer [4], and Newman's [5] articles provide a good overview on the community detection approaches. The community detection approaches include traditional methods, divisive algorithms, and modularity-based methods. As one of the most popular methods, the modularity-based methods have attracted many researchers' attention, with the most characteristic feature of converting the network clustering problem into an optimization problem by maximizing the modularity Q presented by Girvan and Newman [6]. With the increase of network size, calculating the communities with maximal modularity is NP-hard [7]. Therefore, heuristic and intelligent optimization algorithms are often used to tackle the problem. For example, GN algorithm [6] generated candidate solutions by using heuristic operations, such as moving a node to other communities,

switching nodes in different communities, decomposing or merging communities, and then selecting the best community division by calculating Q values with simulated annealing Metropolis criterion. FN [8] algorithm is a community detection algorithm proposed by Newman in 2004; the basic idea is to use greedy optimization algorithm to maximize Q value. BGLL algorithm [9] utilized the network topology and modularity to compute the division of community structures; the time complexity of the algorithm is proved to be linear.

Small-scale communities cannot be detected from some large complex networks by optimizing the modularity, which is the resolution limit [10]. To conquer the limit, a number of modified modularity measurements, such as modularity density [11], kernel k-means (KKM), and ratio cut (RC) [12], are developed and introduced into the objective function, which promotes a group of new detection methods based on multiobjective optimization. Recent multiobjective optimization algorithms for community detection include MOGAnet [13], MOCD [14], and MOEA/D-net [15]. MOGA-net algorithm uses community fitness (CF) and community score (CS) as two objectives to be optimized to solve community detection problem. MOCD algorithm employed PESA-II [16] to optimize the objective functions intra and inter and used two methods ($\max Q$ and $\max D$) to select suitable solutions from Pareto dominant solution set. MOEA/D-net algorithm employed MOEA/D to optimize negative ratio association (NRA) and ratio cut (RC) [15] to find dominant solutions.

The above work shows that the modularity-based intelligent optimization algorithms for community detection attract much attention of researchers. In order to further improve the performance of intelligent optimization algorithms for community detection, the paper proposes a new framework including hybrid evolving strategies and adaptive learning mechanism based on evolutionary algorithm. The work includes two parts. In the first part, the modularity Q is used as the objective function because of its simplicity and easy understanding. A Hybrid Self-adaptive Community Detection Algorithm (HSCDA) based on modularity is put forward. In HSCDA, three different crossover and two different mutation operators for community detection are designed and then combined to form a strategy pool, in which the strategies will be selected probabilistically based on statistical self-adaptive learning framework. Experimental results show that HSCDA is able to achieve competitive modularity compared to other modularity-based algorithms, GN, FN, and BGLL. In the second part, a Multiobjective Community Detection Algorithm (MCDA) is proposed, in which KKM and RC are used as two optimization objectives instead of the modularity. The primary evolving strategy of MCDA is decided by the self-adaptive learning framework in HSCDA. Pareto mechanism is used to preserve the good solutions. Experimental results show that MCDA achieves a better performance compared with HSCDA and other multiobjective based algorithms, MOGA-net, MOCD, and MOEA/D-net.

The rest of this paper is organized as follows: Section 2 gives the problem statements. In Section 3, the proposed algorithms for community detection are presented. In Section 4, the performances of the proposed algorithms are validated on

both computer-generated networks and real world networks. We also compare our algorithms with other approaches. The conclusions are finally summarized in Section 5.

2. Network Community Detection Problem

Assume that a network G is defined as G = (V, E), where V denotes the node set and E denotes the edge set. The topology of the network is usually represented by adjacent matrix $A = (A_{ij})$. The elements in the matrix are 0 or 1. $A_{ij} = 1$ indicates that the nodes i and j are connected, whereas $A_{ij} = 0$ represents the nodes i and j are unconnected.

Community structure is a universal property of many complex networks in real world. The community is the node subset, which has a relatively tight connection between the inner nodes and a relatively sparse connection between the external nodes [17]. Since the concept of connection is not clearly defined, there are many ways to measure community structure.

The modularity *Q* proposed by Girvan and Newman is the most popular measurement [6]. *Q* is defined as follows:

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

where d is the total number of edges in the network, A_{ij} is the adjacent matrix of the network, and k_i is the degree of node i; if the nodes i and j are in the same community, $\delta(i,j)=1$; otherwise it is 0. If the value of Q is bigger than 0, the community structure begins to appear in complex networks. If Q value is greater than 0.3, there is a clear community structure in complex networks. If Q value is close to 1, community structure is more obvious. In the real world complex networks, Q value is usually between 0.3 and 0.7. The advantage of modularity is easy understanding and lower computational cost. The problem of community detection based on modularity is an optimization problem by maximizing modularity Q.

In order to solve problem of limit of modularity resolution [10], Li et al. in [11] introduced a new objective function, the modularity density D, which is defined as

$$D = \sum_{i=1}^{m} \frac{L\left(V_i, V_i\right) - L\left(V_i - \overline{V_i}\right)}{|V_i|},\tag{2}$$

where V_i is the node set of ith community in all m communities, $L(V_i, V_i) = \sum_{i \in V_i, j \in V_i} A_{ij}$, and $|V_i|$ is the node number in V_i . The greater the value of the modularity density D, the more accurate the community found. In order to analyze network topology structure in different resolutions and find community of networks in more detail, the expression of modularity density is improved continuously and then decomposed to several parts to form a multiobjective optimization problem for community detection. One of the

Input: Adjacent matrix A of network G

Parameters: population size (popsize), max generations (gen), crossover probability (pc),

mutation probability (pm), initial probability of adaptive strategy (p)

Output: Optimal solution of the current iteration

Step 1. Initialization

(1.1) Initialize each individual by label propagation mechanism (see Algorithm 2).

(1.2) Calculate objective function Q using formula (1).

Step 2. Self-adaptive learning

For each individual, select a strategy from hybrid evolutionary strategy pool (see Section 3.1.3)

using roulette wheel selection according to the selected probability,

then update the selected probability of each strategy by self-adaptive learning framework (see Section 3.1.4).

Step 3. Local search

Apply hill-climbing search (see Section 3.1.5) to the individual with highest Q value in the current population for local search.

Once a better individual is generated, the new individual will replace the chosen one.

Repeat until no more better individual is get or the number of search reaches the maximum,

then the individual is the current best solution of the population.

Step 4. Stopping criteria:

If (iterations < gen), iterations ++, and go to Step 2; otherwise, stop the algorithm and output.

ALGORITHM 1: The flow of HSCDA.

popular decompositions is KKM [12] and RC [12], which are defined as follows:

KKM =
$$5(n-m) - \sum_{i=1}^{m} \frac{L(V_i, V_i)}{|V_i|}$$
,

$$RC = \sum_{i=1}^{m} \frac{L(V_i, \overline{V_i})}{|V_i|}.$$
(3)

The smaller the KKM value is, the closer the internal group will be, and the smaller the RC is, the sparser the links between nodes of internal and external community will be. Therefore, community detection problem can also be modeled to a multiobjective optimization problem by minimizing KKM and RC.

3. Description of Proposed Method

In this section, the detailed information of HSCDA and MCDA is depicted.

3.1. HSCDA. In order to further improve the solution quality of intelligent optimization algorithms for community detection problems based on modularity, HSCDA is proposed based on evolutionary algorithm. In HSCDA, three different crossover and two different mutation operators for community detection are designed and then combined to form a strategy pool, in which the strategies will be selected probabilistically by roulette wheel selection based on statistical self-adaptive learning framework. The flow of HSCDA is shown in Algorithm 1.

3.1.1. Individual Encoding. A partition Ω of the network G is encoded as an integer string $\mathbf{x} = \{x^1, x^2, \dots, x^n\}$, where \mathbf{x} denotes an individual, n is the number of nodes in the network, and x^i is community label of node i,

 $x^i \in \{1,2,\ldots,n\}$. Nodes with the same community label are considered in the same community. Note that a network of n nodes can be divided into n communities at most; in this case, each node consists of a community, which can be denoted as $\{1,2,\ldots,n\}$. Moreover, there are many different representations corresponding to the same partition. For example, given a network of 4 nodes, $\{2,1,3,2\}$ and $\{1,3,2,1\}$ represent the same partition $\{\{1,4\},\{2\},\{3\}\}\}$; that is, nodes 1 and 4 belong to the first community, node 2 belongs to the second one, and node 3 belongs to the third one. This direct encoding mode can be easily used without knowing the additional information such as the size of community structures in advance.

3.1.2. Population Initialization Algorithm Based on Label Propagation Mechanism. To both reduce the searching space and promote diversity, the paper adopts initialization mechanism based on label propagation [12], which makes full use of prior knowledge of network topology to generate a population that densely connected nodes have a unique label.

Assume that the neighbor set of a node i is $N(i) = (x_1, x_2, ..., x_k)$ and let l(i) be the label of node i. In label propagation mechanism, the label of each node depends on the label with biggest proportion of labels in its neighbor set; it is defined as follows:

$$l(i) = \arg \max_{r} \sum_{j \in N(i)} \delta(l(j), r), \qquad (4)$$

where r represents the community labels of nodes in N(i). If l(j) and r are the same labels, then $\delta(l(j),r)$ equals 1 and otherwise 0. After label propagation, densely connected nodes can be set as the same label quickly. Algorithm 2 shows the flow of initialization algorithm using label propagation.

3.1.3. Hybrid Evolutionary Strategy Pool. In order to enhance the capability of evolution of the algorithm and thus to

```
Input: Population with each node divided into different communities, that is, l(i) = i, i \in \{1, 2, ..., n\}
Output: Population after initialization
                             //for all individuals in population
For m = 1: popsize
  For p = 1:5
                            //the number of propagation iterations is set to 5
     For i = 1:n
                            //for all nodes in the network
      If (N(i) > 1)
           For all node j in N(i)
             l(i) \leftarrow \text{Equation } (4)
           End For
      Else
                                 //node i has only one neighbor j
             l(i) = l(j)
                                //assign the label of neighbor j to i
      End If
     End For
  End For
End For
```

ALGORITHM 2: Algorithm of population initialization.

improve the quality of solution, six different strategies for community detection are designed to make up the hybrid evolutionary strategy pool. Every evolutionary strategy includes crossover and mutation operators. Individual chooses different strategies adaptively and then gradually improves its solution structure.

Given two individuals x_a and x_b , three different crossover and two different mutation operators are designed as follows.

Crossover 1 Is Block Crossover. Two positions of a and b ($1 \le a \le b \le n$) are randomly selected at first. Then, the labels (from 1 to a and from (b+1) to n) are selected from x_a to replace the labels of the same position of a new individual x_c , while the labels in the other position (from (a+1) to b) in x_c are set to the same as in x_b . In the same way, the labels (from 1 to a and from (b+1) to n) are selected from x_b to replace the labels of the same position of a new individual x_d , and the labels in the other position (from (a+1) to b) in x_d are set to the same as in x_a . This process will generate two offspring individuals x_c and x_d .

Crossover 2 Is a Single Point of Double Crossing Crossover. Firstly, randomly select a node called v_i in x_a and mark its label as x_a^i . Then all of nodes with the same label as x_a^i are set to the same label in x_b , thus generating a new individual x_c ; that is, $x_b^k \leftarrow x_a^i$, $\forall k \in \{k \mid x_a^k = x_a^i\}$. Meanwhile, the node v_i with label x_b^i in x_b is found out and then let all of nodes belonging to this community in x_b be set in the same label in x_a , $x_a^k \leftarrow x_b^i$, $\forall k \in \{k \mid x_b^k = x_b^i\}$, thus generating a new individual x_d . This process will generate two offspring individuals x_c and x_d .

Crossover 3 Is a Two-Way Crossing Over [18]. Firstly, randomly select two nodes called v_i and v_j and ensure that their labels x_a^i and x_a^j are different. Then let all of nodes belonging to these two communities in x_a be set as the corresponding communities in x_b to generate a new individual x_c . Meanwhile, the nodes v_i , v_j in x_b are found out with different labels, and make sure that the corresponding nodes in x_a are set as belonging

to these two communities. Thus, the new individual x_d will be generated. This operator is an extended version of Crossover 2 and will also generate two offspring individuals x_c and x_d .

Mutation 1. Firstly, get the community structure according to the labels of nodes of an individual. Secondly, select a node in each community randomly and then change the label of this node into the label of one of its neighbor nodes.

Mutation 2. Firstly, get the community structure according to the labels of nodes of an individual. Secondly, select a node in each community randomly and then change the label of this node into the label of its neighbors which has the highest duplication. If the labels of neighbor node are different from each other, then randomly select a label from neighbor nodes to assign.

Combine the above three crossover and two mutation operators mutually and thus generate the following six evolutionary strategies to form the hybrid evolutionary strategy pool:

```
Strategy 1: Crossover 1 + Mutation 1.
Strategy 2: Crossover 1 + Mutation 2.
Strategy 3: Crossover 2 + Mutation 1.
Strategy 4: Crossover 2 + Mutation 2.
Strategy 5: Crossover 3 + Mutation 1.
Strategy 6: Crossover 3 + Mutation 2.
```

3.1.4. Self-Adaptive Learning Framework. Based on strategy pool, a statistical self-adaptive learning framework is introduced into HSCDA. The individual adaptively chooses the appropriate strategy in different stages of the algorithm depending on the evolution effect of the strategy. In the self-adaptive learning framework, each strategy is given the corresponding probability of being selected. Individual selects evolution strategy by roulette wheel selection.

In particular, each individual i (i = 1, 2, ..., popsize) has a selective probability vector p_i for strategy, $p_i = [p_{i1}, p_{i2}, ..., p_{iM}]$, where p_{ij} means the probability of which

*i*th individual chooses *j*th strategy from all *M* strategies in the hybrid strategy pool. *M* is 6 in the paper.

The difference of the individual before and after evolving by a strategy is used to measure the evolution effect of that strategy, which is defined as follows:

$$diff = \left| \frac{Q_{new} - Q_{old}}{Q_{best}} \right|, \tag{5}$$

where Q_{new} is the modularity of the individual i. Q_{old} represents the modularity of the individual i in last generation. Q_{best} denotes the best individual in current population. Then, the change quantity of the probability is defined as follows:

$$\det a = \frac{\text{rand}}{1 + e^{-\text{diff}}},\tag{6}$$

where rand is a random value in (0,1) used to make a disturbance to avoid learning too fast. Suppose that individual i selects mth strategy by the roulette wheel selection from the strategy pool with the probability of $p_{im}(\text{old})$; then the selective probability of individual i in the next generation will be updated to $p_{im}(\text{new})$, which is calculated as follows when $Q_{\text{new}} - Q_{\text{old}} > 0$:

$$p_{im} (\text{new}) = \frac{p_{im} (\text{old}) + \text{deta}}{p_{im} (\text{old}) + \text{deta} + \sum_{j=1, j \neq m}^{M} p_{ij} (\text{old})}.$$
 (7)

If $Q_{\text{new}} - Q_{\text{old}} \le 0$, $p_{im}(\text{new})$ is calculated as follows:

$$p_{im} (\text{new}) = \begin{cases} \frac{p_{im} (\text{old}) - \text{deta}}{p_{im} (\text{old}) - \text{deta} + \sum_{j=1, j \neq m}^{M} p_{ij} (\text{old})}, & \text{if } p_{im} (\text{old}) - \text{deta} > 0, \\ \frac{(1 - \text{deta}) \times p_{im} (\text{old})}{(1 - \text{deta}) \times p_{im} (\text{old}) + \sum_{j=1, j \neq m}^{M} p_{ij} (\text{old})}, & \text{if } p_{im} (\text{old}) - \text{deta} \le 0. \end{cases}$$
(8)

However for other strategies, the selective probability $p_{ij\neq m}$ (new) should be updated to make sure $\sum_{j=1}^{M} p_{ij}$ (new) = 1. Given $T_1 = p_{im}(\text{old}) + \text{deta} + \sum_{j=1, j\neq m}^{M} p_{ij}(\text{old})$, $T_2 = p_{im}(\text{old}) - \text{deta} + \sum_{j=1, j\neq m}^{M} p_{ij}(\text{old})$, and $T_3 = (1 - \text{deta}) \times p_{im}(\text{old}) + \sum_{j=1, j\neq m}^{M} p_{ij}(\text{old})$, $p_{ij\neq m}$ (new) is calculated as follows:

 $p_{ij\neq m}$ (new)

$$= \begin{cases} \frac{p_{ij\neq m} \text{ (old)}}{T_{1}}, & \text{if } Q_{\text{new}} - Q_{\text{old}} > 0, \\ \frac{p_{ij\neq m} \text{ (old)}}{T_{2}}, & \text{if } Q_{\text{new}} - Q_{\text{old}} \le 0, \ p_{im} \text{ (old)} - \text{deta} > 0, \\ \frac{p_{ij\neq m} \text{ (old)}}{T_{3}}, & \text{if } Q_{\text{new}} - Q_{\text{old}} \le 0, \ p_{im} \text{ (old)} - \text{deta} \le 0. \end{cases}$$
(9)

Individuals in the next generation will make a choice of the evolving strategies according to the updated selective probabilities. Therefore, HSCDA can make the individual adaptively choose the appropriate strategies at different stages.

3.1.5. Local Search. In order to improve convergence speed and alleviate trapping into local optima, the hill-climbing method suggested in [18] is adopted here as a local search mechanism. Hill-climbing method is a kind of optimization method commonly used in local search, which usually starts from an arbitrary solution of current problems and tries to change an element of this solution to find a better solution. Once this change produces a better solution, then the new solution replaces the selected solution. The process is repeated until there is no better solution to be produced or

reaching the stopping criteria. It is worth noting that the hill-climbing method is only for the individual which has the best fitness value, so as to avoid excessive amount of calculation.

3.2. MCDA. Experiments (see Sections 4.2 and 4.3) show that the effect of the community structure detection algorithm based on the optimization of modularity is not good for the real network clustering. In order to further improve the solution quality, MCDA is proposed. In MCDA, strategy 6 with the largest proportion of selection of the best individual in HSCDA is considered as the strategy of MCDA; KKM and RC are set as two objective functions. The reason to adopt single strategy instead of adaptive framework based hybrid strategy pool is that individuals have to compare with each other to calculate chosen probability of evolving strategy in self-adaptive learning framework, while Pareto mechanism in MCDA cannot make a definite decision of which is good or poor between any two individuals. The same reason leads to the fact that the local hill-climbing search cannot be introduced into MCDA directly. The specific flow of MCDA is shown in Algorithm 3.

4. Experimental Results and Analysis

4.1. Normalized Mutual Information. Normalized Mutual Information (NMI) [20] is commonly used to estimate the similarity between the true clustering results and the detected ones. Two vectors, *A* and *B*, are inputted during the process of comparison. *i*th bit of the vector represents the class of *i*th node. The NMI(*A*, *B*) is then defined as follows:

$$NMI = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}C_{ij}\log\left(C_{ij}N/C_{i.}C_{.j}\right)}{\sum_{i=1}^{C_A}C_{i.}\log\left(C_{i.}/N\right) + \sum_{j=1}^{C_B}C_{.j}\log\left(C_{.j}/N\right)}, \quad (10)$$

Input: Adjacent matrix *A* of network *G*

Parameters: population size (*popsize*), max generations (*gen*), crossover probability (*pc*), mutation probability (*pm*) Output: Pareto front solutions.

Step 1. Initialization

- (1.1) Initialize the population with population initialization algorithm based on label propagation mechanism (Algorithm 2)
- (1.2) Calculate individual objective functions KKM and RC with formula (3)
- (1.3) Calculate the rank of each individual

If at least one objective value of individual *p* is better than that of individual *q*,

and all objects of p are not worse than those of q, then p dominates q. This is for each individual division level (rank), and the rank of all non-dominant individuals is defined as 1,

and the other individual's rank plus 1 with the number of individuals who control it.

(1.4) Calculate crowding distance

Calculate the distance between one individual and other individual in the same rank by

the crowding distance calculation method refers to [19].

Step 2. Adopt the evolutionary strategy 6 to generate offspring individuals

Step 3. Pick out the dominant solutions of current generation from the population

The rank of all individuals is calculated first,

then select the individuals whose rank is 1 to construct dominant solutions of the current generation.

Step 4. Using the pruning mechanism to update the population

- (4.1) Combine the dominant solutions with the present population to form a new population
- (4.2) Calculate the rank of each individual and sort them from small to large.
- (4.3) Select *popsize* individuals as the next generation according to the rank.

Step 5. Stopping criteria

If (iterations < gen), iterations ++ and go to Step 2, otherwise, stop the algorithm and output the dominant set of solutions.

ALGORITHM 3: The flow of MCDA.

Table 1: Characteristics of four real world networks.

| Network | Nodes | Edges | True clustering results |
|-------------|-------|-------|-------------------------|
| Karate Club | 34 | 78 | 2 |
| Dolphin | 62 | 159 | 2 |
| Football | 115 | 613 | 12 |
| Polbooks | 105 | 441 | 3 |

where $C_A(C_B)$ is the number of clusters in vector A (B), C is the mixing matrix which consists of vector A and vector B, C_{ij} is the number of elements shared in common by gth classification of vector A and by jth classification of vector B, C_i , $(C_{\cdot j})$ is the sum of elements of C in row i (column j), and N is the number of nodes of the network. The value of NMI(A, B) is in the interval [0,1]. If NMI(A, B) = 1, then A = B. If NMI(A, B) = 0, then A and B are totally different.

4.2. Experimental Results and Analysis of HSCDA. The parameters of HSCDA are set as follows: population size is 100, crossover probability is 0.8, mutation probability is 0.2, the initial selection probabilities of evolving strategies in strategy pool for each individual are set as p = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6], and the maximum number of iterations is 100.

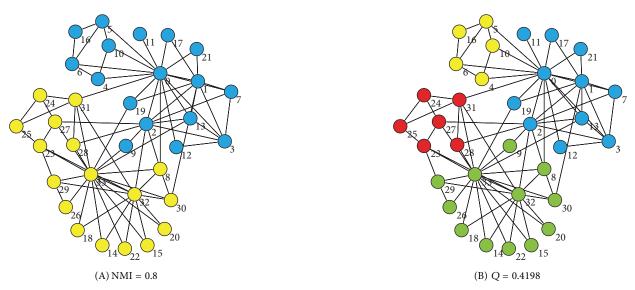
Zachary's Karate Club network [21], Dolphin social network [22], American college Football network [23], and Books on US political network (Polbooks) [24] are commonly used real networks for benchmarking. Characteristics of these four networks are shown in Table 1. For details, please see the related references.

TABLE 2: NMI of HSCDA, GN, FN, and BGLL in four real networks.

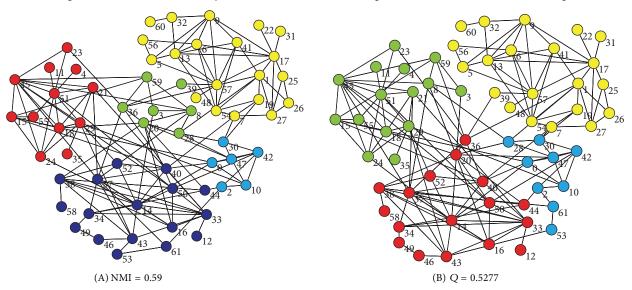
| Algorithm | Karate Club | Dolphin | Football | Polbooks |
|-----------|-------------|---------|----------|----------|
| GN | 0.58 | 0.55 | 0.88 | 0.56 |
| FN | 0.69 | 0.57 | 0.76 | 0.53 |
| BGLL | 0.59 | 0.52 | 0.89 | 0.57 |
| HSCDA | 0.80 | 0.59 | 0.89 | 0.57 |

HSCDA is applied to four real networks, respectively; the average of optimal solutions of HSCDA after running 30 times is recorded. Table 2 lists comparison results between HSCDA and GN, FN, and BGLL algorithm in terms of NMI, where the results of GN, FN, and BGLL are taken from [25]. As seen from the table, the NMIs of HSCDA are superior to other three algorithms except that NMIs are the same as BGLL in Football and Polbooks. Table 3 shows the comparison results of Q values of HSCDA, GN, FN, and BGLL; we can find that Q values obtained from HSCDA are higher than the other three algorithms. This is because adopting hybrid evolution strategies based on self-adaptive learning framework can improve solution quality of HSCDA. Community structures calculated by HSCDA on four real networks are given in Figure 1. Results of Tables 2 and 3 and Figure 1 show that HSCDA is more accurate than GN, FN, and BGLL.

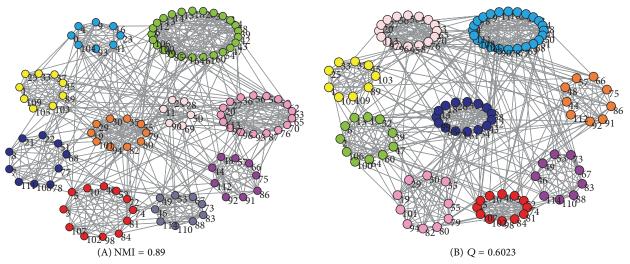
4.3. Analysis of Evolution Effect of Strategies in Self-Adaptive Learning Framework. To analyze the actual evolution effect of evolving strategy in hybrid strategy pool, the selected count of each evolving strategy of the optimal solutions (run 30



(a) Clustering results on Karate Club network by HSCDA. (A) is the structure with highest NMI value; (B) is the structure with highest Q value

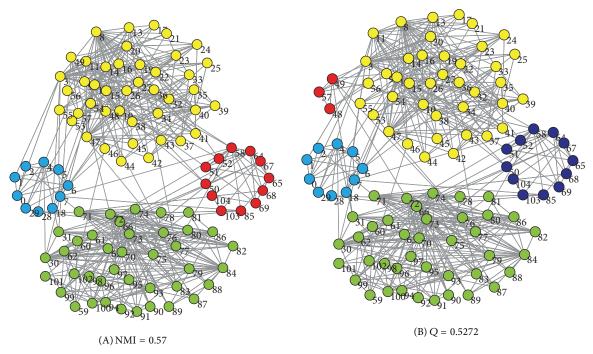


 $(b) \ \ Clustering \ results \ on \ Dolphin \ network \ by \ HSCDA. \ (A) \ is \ the \ structure \ with \ highest \ NMI \ value; \ (B) \ is \ the \ structure \ with \ highest \ Q \ value$



 $(c) \ \ Clustering \ results \ on \ Football \ network \ by \ HSCDA. \ (A) \ is \ the \ structure \ with \ highest \ NMI \ value; \ (B) \ is \ the \ structure \ with \ highest \ Q \ value$

Figure 1: Continued.



(d) Clustering results on Polbooks network by HSCDA. (A) is the structure with highest NMI value; (B) is the structure with highest Q value

FIGURE 1: Clustering results on four real networks by HSCDA.

Table 3: Q value of HSCDA, GN, FN, and BGLL in four real networks.

| Algorithm | Karate Club | Dolphin | Football | Polbooks |
|-----------|-------------|---------|----------|----------|
| GN | 0.4013 | 0.5194 | 0.5996 | 0.5168 |
| FN | 0.3801 | 0.4897 | 0.5773 | 0.5020 |
| BGLL | 0.4188 | 0.5188 | 0.6021 | 0.4986 |
| HSCDA | 0.4198 | 0.5277 | 0.6023 | 0.5272 |

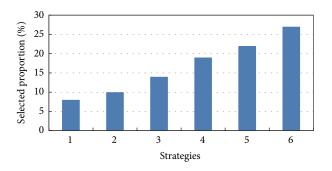


Figure 2: Selected proportion of strategies of the optimal solutions.

times independently) is recorded and shown in Figure 2. As shown in Figure 2, the selected proportion of strategy 6 is the highest in all strategies, which means that the evolution effect of strategy 6 is superior to others when dealing with the community detection problem.

From the results in Tables 2 and 3 and Figure 1, it is shown that HSCDA is superior to other methods based on modularity. However, according to the results in Tables 2 and 3, the improvement of Q is not in accordance with NMI; that is, for Football and Polbooks, Q value of HSCDA is superior to BGLL while NMI is the same as BGLL. The reason of the phenomenon is that O cannot fully disclose the essential of natural group in real networks. To improve the cluster effect, we further propose MCDA. In MCDA, strategy 6 is considered as the strategy of MCDA and KKM and RC are set as two objective functions. The reason to adopt single strategy instead of adaptive framework based hybrid strategy pool is that individuals have to compare with each other to calculate chosen probability of evolving strategy in self-adaptive learning framework, while Pareto mechanism in MCDA cannot make a definite decision of which is good or poor between any two individuals. The same reason leads to the fact that the local hill-climbing search cannot be introduced into MCDA. The experimental results and analysis are detailed in the next section.

4.4. Experimental Results and Analysis of MCDA. The parameters of MCDA are set as follows: population size is 100, crossover probability is 0.9, mutation probability is 0.1, and maximum number of iterations is 100. MCDA and three multiobjective algorithms (MOGA-net [13], MOCD [14], and MOEA/D-net [15]) are compared in experiments on artificial synthetic network and four real world networks, respectively.

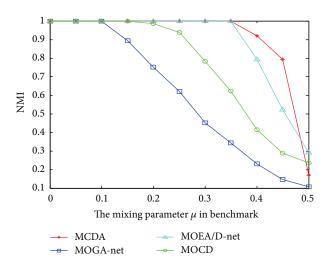


FIGURE 3: Max NMI values averaged over 30 runs for different algorithms.

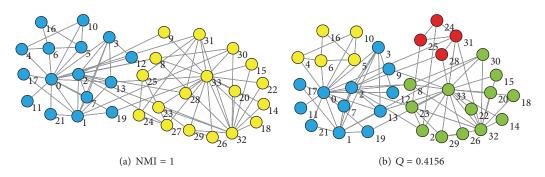


FIGURE 4: Cluster results of MCDA in Karate Club network.

The results show that MCDA has better solution accuracy and obtains true network clusters in several real networks.

4.4.1. Experimental Results and Analysis on Artificial Synthetic Network. In order to compare with other community detection algorithms based on multiobjective optimization, we do experiments on artificial synthetic benchmark network proposed by Lancichinetti et al. [26], which is an extension of classic GN benchmark network proposed by Newman [6]. The network contains 128 nodes which are divided into four communities of 32 nodes each. The average degree of each node is 16. The proportion of outdegree of the node is controlled by mixing parameter. The network becomes vaguer when μ increases, which means that it is harder to figure out the true clusters on this occasion.

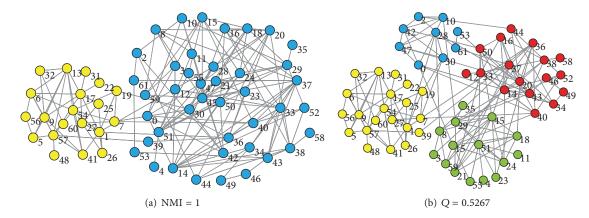
By adjusting values of mixing parameter μ in the synthetic network, 11 networks in which mixing parameter μ changes from 0 to 0.5 with interval 0.05 are generated to test the algorithm. NMI is used to measure the similarity between true network clusters and test results. For each network, we calculate average of the biggest NMI value after the algorithm independently running 30 times. Figure 3 shows the curve of NMI obtained from four different algorithms.

In Figure 3, we found that when $0.1 < \mu < 0.35$, MCDA and MOEA/D-net can find the true network clusters (NMI is 1), while the NMI value of MOGA-net and MOCD declined

obviously. When 0.35 < μ < 0.45, all the algorithms fail to obtain the true clusters, but the NMI of MCDA is still higher than 0.8, which shows that MCDA outperforms other three algorithms when dealing with the vaguer networks. When μ = 0.5, the effect of all algorithms was poor, and it is reasonable since the community structure is fully fuzzy at present. It can be seen from Figure 3 that MCDA has a better performance in most cases (0 < μ < 0.48) compared with MOGA-net, MOCD, and MOEA/D-net, which is the benefit of the good solution space searching ability of strategy 6 for community detection.

4.4.2. Simulation Results and Analysis of Real Networks. MCDA is applied to four real world networks mentioned above. Cluster results with max Q and max NMI are shown from Figures 4 to 7. Figure 4 shows results of Zachary's Karate Club network, Figure 5 shows results of Dolphin social network, results of American college Football network are shown in Figure 6, and results of Books on US politics are shown in Figure 7.

From Figure 4(a), it is clear that MCDA can successfully detect the true community structures (corresponding to NMI = 1). Figure 4(b) shows the community structure corresponding to highest Q value. It is obvious that Figure 4(b) is the subgraph of Figure 4(a).



 $\label{eq:figure 5:Cluster results} Figure \ 5: Cluster \ results \ of \ MCDA \ in \ Dolphin \ network.$

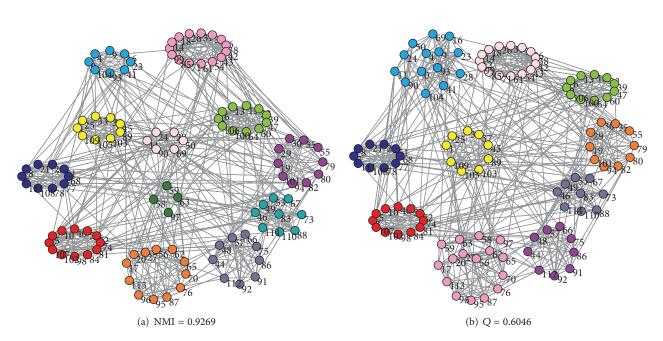
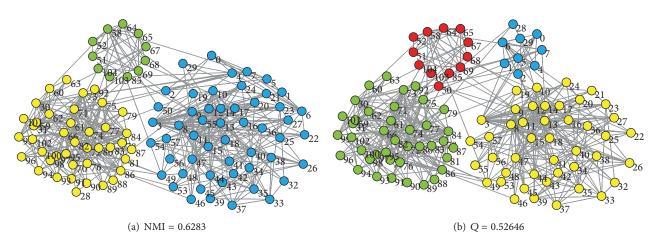


Figure 6: Cluster results of MCDA in Football network.



 $\label{thm:figure 7} \textit{Figure 7: Clustering results of MCDA in Polbooks network.}$

Figure 5(a) shows that MCDA obtains the true community structures of Dolphin social network (NMI = 1). Figure 5(b) shows MCDA divides the structure on the right part in Figure 5(a) into 3 smaller communities. Thus, from optimizing modularity Q point of view, MCDA is also effective for detecting the community structures of Dolphin network without wrong clustering.

Some nodes in Football network are not connected with nodes in the same community, while the connection between nodes of this community and nodes of other communities is more close. When the network is in the real clustering, the modularity Q is -0.0239, which is much less than Q value obtained by the algorithm. It shows that the true clusters are not completely complying with network community cluster rule. Because of the complicated structure, it is difficult to completely detect its real cluster. According to the cluster results from Figure 6(b) with max Q, MCDA obtains 10 clusters. We observed that some nodes like 12, 25, 29, 37, 43, 51, 59, 60, 64, 70, 81, 83, 91, 98, and 111 are misplaced. Figure 6(a) shows community structures detected by MCDA with NMI = 0.9269; it still has a good reference value because of the high NMI.

Similar to Football network, Books on US politics network itself shows high complexity. From the comparison of Figures 7(a) and 7(b), although part of nodes is misplaced and real clusters cannot be completely detected, it can still make NMI be 0.6283 and Q be 0.5264, which is meaningful in terms of solution precision.

5. Conclusion

To further improve the solution quality of intelligent optimization algorithms for community detection, HSCDA and MCDA are proposed based on evolutionary algorithm, respectively. In HSCDA, Q is set as the objective function and six different evolution strategies are designed to construct hybrid evolution strategy pool. Evolution strategy is chosen according to the probability through roulette wheel selection based on statistical self-adaptive learning framework. In MCDA, KKM and RC are set as the two objective functions; strategy 6 which has the largest proportion of selection of the best individual in HSCDA is set as the main evolution strategy and the dominant solution set is kept with Pareto mechanism. Experiments show that HSCDA has higher solution quality compared with other community detection algorithms which use Q as the objective function (such as GN, FN, and BGLL). Compared with HSCDA, MCDA can obtain true structure of some of the real world networks and achieves competitive results compared with other multiobjective community detection algorithms (such as MOGA-net, MOCD, and MOEA/D-net).

Conflict of Interests

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

Acknowledgments

This paper was supported by China Postdoctoral Science Foundation funded project (2015M571790) and NUPTSF (Grant nos. NY213047, NY213050, NY214102, and NY214098).

References

- [1] M. E. J. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 2, pp. 404–409, 2001.
- [2] S. Fortunato, "Community detection in graphs," *Physics Reports-Review Section of Physics Letters*, vol. 486, no. 3–5, pp. 75–174, 2010.
- [3] G. W. Flake, S. Lawrence, C. Lee Giles, and F. M. Coetzee, "Self-organization and identification of web communities," *Computer*, vol. 35, no. 3, pp. 66–71, 2002.
- [4] S. E. Schaeffer, "Graph clustering," Computer Science Review, vol. 1, no. 1, pp. 27–64, 2007.
- [5] M. E. J. Newman, "Communities, modules and large-scale structure in networks," *Nature Physics*, vol. 8, no. 1, pp. 25–31, 2012.
- [6] M. Girvan and M. E. J. 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.
- [7] U. Brandes, D. Delling, M. Gaertler et al., "On modularity clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 172–188, 2008.
- [8] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, Article ID 066133, 2004.
- [9] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P1008, 2008.
- [10] 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.
- [11] 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.
- [12] 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, 2014.
- [13] 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.
- [14] C. Shi, Z. Yan, Y. Cai, and B. Wu, "Multi-objective community detection in complex networks," *Applied Soft Computing Journal*, vol. 12, no. 2, pp. 850–859, 2012.
- [15] M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," *Physica A: Statistical Mechanics and Its Applications*, vol. 391, no. 15, pp. 4050–4060, 2012.
- [16] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: region-based selection in evolutionary multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '01)*, San Francisco, Calif, USA, July 2001.

- [17] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, 2004.
- [18] M. Gong, B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks," *Physical Review E*, vol. 84, no. 5, Article ID 056101, 2011.
- [19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [20] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, Article ID P09008, 2005.
- [21] http://networkdata.ics.uci.edu/data/karate/karate.zip.
- [22] http://networkdata.ics.uci.edu/data/dolphins/dolphins.zip.
- [23] http://networkdata.ics.uci.edu/data/football/football.zip.
- [24] http://networkdata.ics.uci.edu/data/polbooks/polbooks.zip.
- [25] K. Deng, J. Zhang, and J. Yang, "Community detection in complex networks using an improved genetic algorithm," *Journal of Harbin Engineering University*, vol. 34, no. 11, pp. 1438–1444, 2013.
- [26] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, Article ID 046110, 2008.

















Submit your manuscripts at http://www.hindawi.com











Journal of Discrete Mathematics











