

The Improved Estimation of Distribution Algorithms for Community Detection

Zhou Ben-Da^{1,2}

¹. School of Finance & Mathematics, West Anhui University, Lu'an Anhui, China

². Financial Risk Intelligent Control and Prevention Institute, West Anhui University, Lu'an, Anhui, China
e-mail: bendazhou@163.com

Zhao Pan^{1,2}

¹. School of Finance & Mathematics, West Anhui University, Lu'an Anhui, China

². Financial Risk Intelligent Control and Prevention Institute, West Anhui University, Lu'an, Anhui, China
e-mail: zhaopan@163.com

Zhang JinBo^{1,2}

¹. School of Finance & Mathematics, West Anhui University, Lu'an Anhui, China

². Financial Risk Intelligent Control and Prevention Institute, West Anhui University, Lu'an, Anhui, China
e-mail: jinbozhang@163.com

Abstract—Based on the analysis of local monotonic of modularity function, this paper designs a fast and effective mutation operator, and then proposes an improved Estimation of Distribution Algorithm (EDA) for solving community detection problem. The proposed algorithm is tested on basic network and big scale complex network. Experimental results show that this algorithm can get 0.419 8 for the average Q function while running 100 times, has better performance than Girvan-Newman(GN) algorithm, Fast Newman (FN) algorithm and Tasgin Genetic Algorithm (TGA).

Keywords—community detection; estimation of distribution algorithms(EDA); complex network; mutation with local search(LSM)

I. INTRODUCTION

At present, studies of the basic statistical properties of complex networks have attracted the attention of many researchers from different fields. Thus, complex network analysis has become one of the most important interdisciplinary research areas^[1-4]. The characteristics of nodes that belong to a specific class are the same in a network, which is also the case for the clustering characteristics, where nodes are more closely connected with nodes in the same class, whereas connections with different types of nodes are rare. The process of identifying each class in a network is known as network clustering (or network community detection). The study of this problem has major value for applications as well as theoretical significance because it has been used widely for Web community mining, protein function analyses, the prediction of metabolic pathways, and other uses^[5-6]. In general, the evaluation function used to measure the network community structure is the module function proposed by Newman^[6] in 2004 (also known as the Q function). The use of the Q function as an objective function method for detecting the community structure of networks has increased since its proposal, although the maximum of the Q function is NP-complete^[7]. However, genetic algorithms (GAs), simulated annealing

(SA), and other intelligent optimisation methods have been applied successfully to solve this problem^[8-11].

Estimation of distribution algorithms (EDAs) are a new type of random optimisation algorithm, which are used increasingly in the field of evolutionary computation^[12-14]. These methods combine GA theory and statistical learning rules and make full use of the global information related to a population to establish a probability distribution model for an individual space. The probability models are learned and sampled to generate a new solution, which facilitates population evolution^[14]. EDAs can avoid certain GA operations such as crossover and mutation, which can improve their problem-solving capacity. However, EDAs cannot use local information directly during the evolution process because they lack a control mechanism for determining the local optimal solution. In this study, we propose a new distributed estimation algorithm to solve the community detection problem (EDA with LSM, LSMEDA) based on the analysis of the monotonicity of the Q function, where the Q function is used as a community detection quality metric. Further, gene mutation in the GA depends on the monotonicity of a Q function designed for local search and mutation with the EDA, which improves the updating properties of the probability model.

II. THE Q FUNCTION AND ITS PROPERTIES

The design of the Q function is based on the observation that as the network community structure becomes more obvious, the difference between it and random networks increases^[7-11]. The specific details of the Q function are described as follows.

Assume that we are given the undirected and unweighted network $N(V, E)$, the vertex set of V , and the set of edges of E , where V is clustered (divided) into a number of communities. If we suppose that $R(i)$ are the community tags of node i , and the set $Cr(i)$ is the community node set that i belongs to, then the Q function is expressed as

$$Q = \frac{1}{2m} \sum_{i,j} ((A_{ij} - \frac{k_i k_j}{2m}) \times \delta(r(i), r(j))), \quad (1)$$

where the adjacency matrix of the network is $A = (A_{ij})_{n \times n}$,

$$\delta(u, v) = \begin{cases} 1 & u = v, \text{ the degree of node } i \text{ is } k_i, \text{ and the total} \\ 0 & u \neq v \end{cases}$$

number of edges is m .

The Q function is the difference between the number of edges in the actual network within the community and a completely randomly connected community with the expected number.

In the following section, we start from the local viewpoint to analyze the Q function; thus, we simplify equation (1) as follows:

$$\begin{cases} Q = \frac{1}{2m} \sum_i f(i) \\ f(i) = \sum_{j \in C_{r(i)}} (A_{ij} - \frac{k_i k_j}{2m}) \end{cases} \quad (2)$$

The Q function can be expressed by the function $f(i)$ of all nodes. Obviously, $f(i)$ can be understood as follows: the difference between the actual number of connections within the community and the expected number of connections in a randomly connected community. Thus, from the local viewpoint, $f(i)$ can measure the community structure of the network. The following discussion considers the related properties of $f(i)$.

Property 1: For any i that belongs to V , $f(i)$ is only related to the community to which node i belongs.

Proof: For $f(i) = \sum_{j \in C_{r(i)}} (A_{ij} - \frac{k_i k_j}{2m})$, this formula is the sum

of j , which is apparently only related to the nodes.

Property 2: The value of Q function increases monotonically with $f(i)$.

Proof: Assume that the network $N=(V, E)$, $C=\{c_{r(1)}, c_{r(2)}, \dots, c_{r(k)}\}$ is a community division. $c_{r(i)}$ contains the nodes labelled with $r(i)$. For any i that belongs to V , the community label is $r(i)$, and the corresponding community is $c_{r(i)}$. Suppose that only the community label of i changes from $r(i)$ to $r(j)$ ($r(i) \neq r(j)$), whereas the other nodes do not change, which forms a new community structure C' . Thus, C' contains $c'_{r(i)} = c_{r(i)} \setminus \{i\}$ and $c'_{r(j)} = c_{r(j)} \cup \{i\}$. By combining function f in formula (2), the definition of the Q function, and **Property 1**, the f function value will be the corresponding changes, i.e. the set $c = c_{r(i)} \cup c_{r(j)}$, $c' = c'_{r(i)} \cup c'_{r(j)}$. Of course, we have $c = c' = c'_{r(i)} \cup c_{r(j)} \cup \{i\}$; hence, we can discuss the changes in the value of function f in c .

1) For any s in $c'_{r(i)}$, there is:

$$\Delta f_s = \sum_{t \in c'_{r(i)}} (A_{st} - \frac{k_s k_t}{2m}) - \sum_{t \in c_{r(i)}} (A_{st} - \frac{k_s k_t}{2m})$$

for $c'_{r(i)} = c_{r(i)} \setminus \{i\}$; thus, $\Delta f_s = -(A_{si} - \frac{k_s k_i}{2m})$. For any q in $c_{r(i)}$,

there is:

$$\Delta f_q = \sum_{t \in c'_{r(j)}} (A_{qt} - \frac{k_q k_t}{2m}) - \sum_{t \in c_{r(j)}} (A_{qt} - \frac{k_q k_t}{2m})$$

for $c'_{r(j)} = c_{r(j)} \cup \{i\}$; thus, $\Delta f_q = (A_{qi} - \frac{k_q k_i}{2m})$.

For node i : node i that belonged to $c_{r(i)}$ now belongs to $c'_{r(j)}$;

thus, $\Delta f_i = \sum_{p \in c'_{r(j)}} (A_{ip} - \frac{k_i k_p}{2m}) - \sum_{p \in c_{r(i)}} (A_{ip} - \frac{k_i k_p}{2m})$ and, as

defined by formula (2), we have:

$$\begin{aligned} \Delta Q &= \frac{1}{2m} (\sum_{s \in c'_{r(i)}} \Delta f_s + \sum_{q \in c_{r(j)}} \Delta f_q + \Delta f_i) \\ &= \frac{1}{2m} (\sum_{q \in c_{r(j)}} (A_{qi} - \frac{k_q k_i}{2m}) - \sum_{s \in c'_{r(i)}} (A_{si} - \frac{k_s k_i}{2m}) + \Delta f_i). \end{aligned}$$

Because $c_{r(j)} \cup \{i\} = c'_{r(j)}$ and $c'_{r(i)} \cup \{i\} = c_{r(i)}$, we have:

$$\sum_{q \in c_{r(j)}} (A_{qi} - \frac{k_q k_i}{2m}) + (A_{ii} - \frac{k_i k_i}{2m}) = \sum_{q \in c'_{r(j)}} (A_{qi} - \frac{k_q k_i}{2m}) \quad \text{In}$$

addition,

$$\sum_{s \in c'_{r(i)}} (A_{si} - \frac{k_s k_i}{2m}) + (A_{ii} - \frac{k_i k_i}{2m}) = \sum_{s \in c_{r(i)}} (A_{si} - \frac{k_s k_i}{2m}); \text{ thus,}$$

$$\begin{aligned} \Delta Q &= \frac{1}{2m} (\sum_{q \in c_{r(j)}} (A_{qi} - \frac{k_q k_i}{2m}) + (A_{ii} - \frac{k_i k_i}{2m}) \\ &\quad - \sum_{s \in c'_{r(i)}} (A_{si} - \frac{k_s k_i}{2m}) - (A_{ii} - \frac{k_i k_i}{2m}) + \Delta f_i) \\ &= \frac{1}{2m} (\sum_{q \in c_{r(j)}} (A_{qi} - \frac{k_q k_i}{2m}) - \sum_{s \in c_{r(i)}} (A_{si} - \frac{k_s k_i}{2m}) + \Delta f_i) \\ &= \frac{1}{2m} (\Delta f_i + \Delta f_i) = \frac{\Delta f_i}{m} \end{aligned}$$

Thus, the value of the Q function increases monotonically with f_i .

Note: Property 2 illustrates that the other node labels in the network are invariant. If a change in the node i label increases the value of $f(i)$, the Q function of the entire network will also increase in value.

III. ESTIMATION OF DISTRIBUTION ALGORITHMS

A. Individual Coding and Fitness Function Design

The methods used to encode community detection algorithms generally involve string encoding and graph-based encoding^[15]. Compared with graph-based encoding, the application of string coding to the network community structure is more intuitive and efficient^[15]. Let n be the number of vertices in N and suppose that $R=\{r(1), r(2), \dots, r(n)\}$ is a division of the network community (chromosome), where $r(i)$ is the vertex i that corresponds to the community tags (an integer). For vertices i and j in the network, $r(i) = r(j)$ indicates that i and j are in the same community, otherwise i and j are located in different communities.

EDAs select the best individual on the basis of a fitness function, where the network module function can quantitatively evaluate the pros and cons of the network community structure. In the present case, the Q function is

used as the fitness function to evaluate the merits of an individual.

B. Probability Matrix and Its Update Method

The probability models of EDAs are as follows: independent of the variable type (PBIL, UMDA, and CGA), bivariate correlation type (MIMIC and BMDA), and multivariate correlation type (ECGA, FDA and BOA) [14]. In the present study, we consider a large-scale network; therefore, we use PBIL to represent the solution space. However, traditional PBIL uses binary code; thus, we designed a binary matrix for the community label matrix with n nodes in the network, and we set the maximum value of the network community tag as K . The community label matrix of individuals is defined as $F=(f_{ij})_{n \times K}$, where $f_{ij} \in \{0,1\}$ and $\sum_j f_{ij} = 1$. Thus, each node can only be assigned one tag (this can be relaxed, which can solve the problem of overlapping community detection). $f_{ij} = 1$ indicates that the community label of node i is j , and we define its corresponding probability matrix as $P=(p_{ij})_{n \times K}$ for $p_{ij} \in [0,1]$, where p_{ij} indicates the probability of the community tags of node i for j , and $\sum_j p_{ij} = 1$. The update process for the probability matrix proceeds as follows (i.e. the update process for the matrix of individual probabilities).

Suppose that the probability matrix of an individual in generation t is $P(t)=(p_{ij}(t))_{n \times K}$. Its corresponding community label matrix is $F(t)=(f_{ij}(t))_{n \times K}$. Given that m individuals are selected for updating, the probability $p_{ij}(t+1)$ in generation $(t+1)$ that expresses the community tags of node i for j is as follows:

$$p_{ij}(t+1) = (1-\alpha)p_{ij}(t) + \alpha \frac{\sum_k f_{ij}^k(t)}{m} \quad (3)$$

where $f_{ij}^k(t)$ represents the community tag of the k -th individual node i in generation t 's that needs to be updated for an individual j . Thus, $\sum_k f_{ij}^k(t)$ represents the number of nodes i for which the community tag is j in m individuals, and $\alpha \in (0,1)$ is the speed of learning, where a higher value of α indicates that the algorithm updates and learns more rapidly.

C. Mutation Operator

An advantage of an EDA is that it describes the search space using a probability model, which is a feasible solution to the global distribution of statistical information^[16]. However, EDAs lack the capacity for the effective use of the local optimal solution during each generation, while there is also no mechanism for direct control of the local optimal solution^[6]. Therefore, based on **Property 2**, we designed a mutation operator to overcome the latter deficiency.

Let R be the variation in a chromosome assuming that node i in R was selected for the mutation operation, and $v(i)$ is a label set of i 's neighbour node. Based on **Property 2**, it is not necessary to consider all of the nodes in R ; thus, we select from $v(i)$ to change i 's tag and maximise the function $f(i)$. The detailed algorithm is as follows.

Algorithm 1. Pseudocode of the mutation operator **LSM** (R is the individual variation, and b is the gene mutation probability).

LSM procedure

```

Global  $R, \beta$ 
Begin
1 For  $i = 1:n$ 
2   If  $\text{rand}() < \beta$ 
3      $v(i) \leftarrow$  The complete neighbour node set of  $i$ 
4      $\text{label}(i) \leftarrow$  Corresponding label set of a node in  $v(i)$ 
5      $\text{maxval} \leftarrow$  Current division that corresponds to  $f_i$ 
6     For each  $r \in \text{label}(i)$ 
7        $f_i \leftarrow$  Corresponding value of  $f$  when changing the
label of node  $i$  from  $r(i)$  to  $r$ 
8       If  $f_i > \text{maxval}$ 
9          $f_i \leftarrow \text{maxval}$ 
10         $\text{label} \leftarrow r$ 
11      End If
12    End For
13     $r(i) \leftarrow \text{label}$ 
14  End If
15 End For

```

Clearly, after each mutation operation, the value of $f(i)$ does not decrease; hence, the increase in the value of the Q function can improve the fitness in an effective manner. Indeed, the mutation operator can be viewed as the local search operation of the EDA. Furthermore, if we specify the gene mutation probability β , a higher value of β will increase the variability of the node in R . Thus, $\beta=1$ is the equivalent of a local search for chromosome R . This mutation operator can greatly improve the solution space for this algorithm, which ensures the diversity of the population and individuals.

In the present study, β takes the reciprocal of the number of different chromosomes in the population, which is small during the initial stage of evolution, but it increases over time to facilitate adaptive adjustment.

IV. EDA FOR SOLVING THE COMMUNITY DETECTION PROBLEM AND ITS PERFORMANCE ANALYSIS

A. Improved EDA for Solving the Community Detection Problem

Algorithm 2. LSMEDA algorithm.

```

LSMEDA procedure
Begin
1   $D_l^0 \leftarrow$  Randomly generate  $M$  individuals as the initial
groups
2  Calculate the fitness values of  $M$  individuals
3  If the end condition is not satisfied
4     $D_l^s \leftarrow$  Select  $m = \gamma M$  ( $0 < \gamma < 1$ ) individuals as the
dominant groups
5     $D_l^{s'} \leftarrow$  Randomly select  $m' = \lambda M$  ( $0 < \lambda < 1$ )
individuals from  $m$  for the LSM operation

```

6 $D_l'' \leftarrow D_l'$ and the remaining individuals of D_l' form a new dominant group
7 According to the probability matrix update rule, (3) updates the probability matrix D_l'' and retains the best individual
9 Based on the probability matrix M times sampling, go to step 3 to obtain a new generation of the population
10 Else
11 over
12 End If

Note: The probability matrix is obtained with a uniform distribution during initialisation, i.e. $p_{ij}=1/K$. Thus, the probabilities of assigning nodes different labels are equal at the beginning of the algorithm.

First, the LSMEDA algorithm generates the initial population before iteratively performing selection, performing local search variation LSM, learning to build the probability model, executing a sampling operation that corresponds to the dominant individual, and producing the next generation of groups. The probability model is established to implement the global search capacity. The LSM corresponds to a local search function with a local optimisation mechanism. The operator is selected using a 'survival of the fittest' mechanism, where the fittest chromosomes are used as the population in the next generation. Thus, the LSMEDA algorithm performs global and local search in an effective manner.

B. Algorithm Performance Analysis Units

Conclusion 1: The LSM operator does not decrease the value of the fitness function of the variant chromosome Q .

Proof: According to the algorithm flow, the variant gene i of chromosome R is operated on by the LSM, and the tag aims to maximise $f(i)$, whereas the other node label does not change. According to **Property 2**, this conclusion is correct.

Conclusion 2: The time complexity of the LSM operator is $O(nc)$.

Proof: If the network has n nodes, the average degree of the nodes in the network is k , the average community scale of R is C , and complex networks can be represented as a sparse graph. According to the algorithm, line 6 ensures that the algorithm has the highest time complexity (i.e. take all of the neighbour tags of the variation vertex and compute the function values at the same time). During actual operation, the neighbour tag of node i 's tag may be repeated, so the average time required to calculate the value of $f(i)$ must be less than or equal to K . According to **Property 1**, only the community information is required to calculate $f(i)$; thus, the average time required to calculate $f(i)$ is c ; therefore, the time complexity of the algorithm does not exceed $O(\beta knc)$ because the number of variant nodes is approximately βn . k and β are constants, so the complexity of the algorithm should be $O(nc)$.

Conclusion 3: The time complexity of the LSMEDA algorithm is also $O(nc)$.

Proof: If the network has n nodes, the average degree of the nodes in the network is k . Let c be the average community scale of the variant chromosome R , and it

performs algebra with L . The highest time complexity in this algorithm is **line 5**, while the runtime of the other steps is less than or equal to $O(n)$. Line 5 in the algorithm has λLM times variation at most; thus, the execution time of the mutation operator is $O(cn)$. This means that the time required for line 5 does not exceed $O(\lambda LMcn)$, when k is a constant, the population scale is M , the executive algebra is L , and the parameter g is also regarded as a constant. Therefore, the algorithm's time complexity is $O(cn)$.

V. EXPERIMENTAL CONCLUSION AND ANALYSIS

In our quantitative analysis of the LSMEDA algorithm, we used a benchmark network and a real large-scale network for testing and verification. The experimental environment used to implement the algorithm was as follows: an Intel(R) Core(TM) M450 processor running at 2.40 GHz, 6.00 GB of memory, a 320-GB hard disk, Microsoft Windows 7 64-bit operating system, and the Matlab 7.10 programming environment.

The algorithm parameters were: maximum iterative algebra $L = 100$, population size $PopSize = 500$, mutation probability $\beta = 0.6$, learning rate $\alpha = 0.1$, truncation selection parameter $\gamma = 0.5$, and mutation selection parameter $\lambda = 0.5$ (50% of the fittest population was selected as the dominant group, and 50% of the dominant population's mutations were selected).

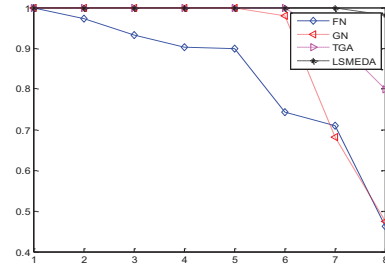


Figure 1. Community detection accuracy comparison

A. Computer-generated Network

In 2002, Newman et al. proposed a random network model $RN(a, s, d, zout)$ to test the performance of a detection algorithm in a network community^[15]. The community structures are known in the model, a represents the size of the network community, s represents the number of nodes in each community, d represents the average node degree, and $zout$ represents the average number of edges between nodes inside the community and outside the community. A previous study^[17] proposed the use of standard normalised mutual information (NMI) to evaluate the accuracy of the clustering effects based on information theory and demonstrated that NMI was fair and reasonable. In the present study, we used the NMI method to evaluate the clustering accuracy of different network community detection algorithms by comparing LSMEDA with popular algorithms such as GN^[13], FN^[18], and Tasgin's GA (TGA)^[8]. The results are shown in Figure 1. The random network parameter used in the diagram is $RN(4, 32, 16, zout)$. If $zout$

is larger, the detection algorithm finds it more difficult to determine the fuzzy network's corresponding community structure.

If $z_{out} > 8$, i.e. the average number of edges between communities is greater than the average number of edges within the community, we considered that the network did not have a community structure^[3]. In the figure, the y-axis represents the clustering accuracy, and the value on the x-axis represents the value of z_{out} . Each data point on the curve represents the average accuracy of the algorithm based on 100 runs. The figure shows that the clustering accuracy of the LSMEDA algorithm was much higher than that of GN, FN, and TGA as z_{out} increased (as the network community structure became increasingly fuzzy). Thus, LSMEDA was obviously the best method. In addition, even when $z_{out} = 8$ (the average number of edges between communities was equal to the average number of edges within the community), the LSMEDA algorithm could still detect the community structure, whereas the GN, FN, and TGA algorithms were ineffective.

B. Real complex networks

In this study, four widely used real networks^[19-21] were employed as the test dataset to further verify the validity and advantages of LSMEDA. TABLE I provides brief descriptions of the networks.

TABLE I. NETWORK DATA INFORMATION

Network name	Number of vertices	Number of edges in the network	Network description
Karate	34	78	Zachary's karate club
Dolphin	62	160	Dolphin social network
Polbooks	105	441	Books about US politics
Jazz	198	5484	Jazz musicians network

Using the real complex networks in TABLE I, we compare the performance of the LSMEDA, GN, FN, and TGA algorithms. TABLE II shows the average Q functions after executing the algorithms 100 times.

TABLE II. COMPARISON OF LSMEDA, GN, FN, AND TGA IN A REAL NETWORK

Algorithm	Karate	Dolphin	Polbooks	Jazz
GN	0.4013	0.4706	0.5168	0.4051
FN	0.2528	0.3715	0.5020	0.4030
TGA	0.4039	0.5241	0.5245	0.4406
LSMEDA	0.4198	0.5280	0.5272	0.4449

Table II shows that the LSMEDA algorithm performed better than the other algorithms when detecting communities in real complex networks. For the karate network, the real community structure corresponded to a Q value of 0.3715, but after executing our algorithm 100 times, we obtained an average Q function value of 0.4198.

VI. CONCLUSIONS

In this study, we used the Q function as an objective function, employed string encoding, and designed a community partition matrix and a community tags probability matrix. Based on an analysis of the monotonic

properties of the Q function, we designed a local search variant operator and proposed a new EDA and LSMEDA for solving the community detection problem. The experiments showed that the LSMEDA algorithm had a faster convergence speed and a strong search capacity in large-scale complex networks, but it could still obtain high quality results in a rapid manner. In our future work, we intend to analyze and improve the probabilistic model used by the LSMEDA algorithm and apply it to the analysis of gene regulation networks or Web community mining research.

ACKNOWLEDGMENT

Research partially supported by the Natural Science Foundation of Anhui Province (Project No. 1808085MG224), the Natural Science Key Foundation of the Education Department of Anhui Province (Project No. KJ2017A402), the Soft Science Foundation of Anhui Province (Project No. 1706a02020037), and the Natural Science Key Foundation of West Anhui University (Project No. WXZR201619).

REFERENCES

- [1] Watts D J, Strogatz S H. Collective Dynamics of Small-world Networks[J]. Nature, 1998, 393(6684): 440-442
- [2] Adamic L A, Huberman B A, Barabasi A L, et al. Power-law Distribution of the World Wide Web[J]. Science, 2000, 287(5461).
- [3] Girvan M, Newman M E J. Community Structure in Social and Biological Networks[J]. Proceedings of National Academy of Sciences, 2002, 99(12): 7821-7826.
- [4] Yan G, Chen G, Lv J, et al. Synchronization Performance of Complex Oscillator Networks[J]. Physical Review E, 2009, 80(5).
- [5] Fortunato S. Community Detection in Graphs[J]. Physics Reports, 2010, 486(3-5): 75-174.
- [6] Newman M E J, Girvan M. Finding and Evaluating Community Structure in Networks[J]. Physical Review E, 2004, 69(2).
- [7] Brandes U, Delling D, et al [EB/OL]. (2015-12-21). http://www.researchgate.net/publication/2176834_Maximizing_Modularity_is_hard.
- [8] Tasgin M, Herdagdelen A, Bingol H. Community Detection in Complex Networks Using Genetic Algorithms[EB/OL]. (2015-12-21). <http://arxiv.org/abs/cond-mat/0604419>.
- [9] Li Shuzhuo, Chen Yinghui, Du Haifeng, et al. A Genetic Algorithm with Local Search Strategy for Improved Detection of Community Structure[J]. Complexity, 2010, 15(4): 53-60.
- [10] Pizzuti C. Community Detection in Social Networks with Genetic Algorithms[C]//Proc. of the 10th Annual Conference on Genetic and Evolutionary Computation. New York, USA: ACM Press, 2008.
- [11] Jin D, He D, Liu D, Baquero C. Genetic Algorithm with Local Search for Community Mining in Complex Networks[C]//Proc. of the 22nd IEEE International Conference on Tools with Artificial Intelligence. Arras, France: IEEE Press, 2010.
- [12] Hauschild M, Pelikan M, Lima C, et al. Analyzing Probabilistic Models in Hierarchical BOA on Traps and Spin Glasses[C]//Proc. of GECCO'07. London, UK: IEEE Press, 2007.
- [13] Liu Ming-Fang, Research on Integer Programming Based on Estimation of Distribution Algorithm[D], Wuhan: Wuhan University of Technology, 2008
- [14] ZHOU Shu-De, SUN Zeng-Qi. A Survey on Estimation of Distribution Algorithms[J]. ACTA AUTOMATICA SINICA, 2007, 33(2): 113-124.
- [15] JIN D, LIU Jie. Genetic Algorithm with Local Search for Community Detection in Large-scale Complex Networks[J]. ACTA AUTOMATICA SINICA, 37(7): 873-882.

- [16] [16] He Xiao-Dong. Research on Intelligent Algorithms for Network Community Mining[D], ChangChun: JiLin University,2010
- [17] Zhu Wen-Qiang. Application of Rough Set and Ant Colony Algorithm on Community Discovery[J], SuZhou:SooChow University,2011
- [18] ZHOU Ben-da, YAO Hong-liang, LI Guo-cheng. Estimation of Distribution Algorithm for Solving Clustering Problem[J], Computer Engineering, 2011,37(22): 191-195.
- [19] Danon L, Diaz G A, Duch J, et al. Comparing Community Structure Identification[J]. Journal of Statistical Mechanics: Theory and Experiment, 2005, 2005(9).