http://doi.org/10.7566/JPSJ.85.124005

Estimation of Distribution Algorithm with Local Sampling Strategy for Community Detection in Complex Networks

Fahong Yu[1]*, Wenping Li[1], Feng He[1], Bolin Yu[2], Xiaoyun Xia[1], and Longhua Ma[3]

[1]*College of Mathematics and Information Engineering, Jiaxing University, Zhejiang 434023, China*
[2]*School of Electronics and Communication, Shenzhen Institute of Information Technology, Shenzhen 518172, China*
[3]*Ningbo Institute of Technology, Zhejiang University, Ningbo 315100, China*

(Received August 1, 2016; accepted October 3, 2016; published online November 18, 2016)

It is important to discover the potential community structure for analyzing complex networks. In this paper, an estimation of distribution algorithm with local sampling strategy for community detection in complex networks is presented to optimize the modularity density function. In the proposed algorithm, the evolution probability model is built according to eminent individuals selected by simulated annealing mechanism and a local sampling strategy based on a local similarity model is adopted to improve both the speed and the accuracy for detecting community structure in complex networks. At the same time, a more general version of the criterion function with a tunable parameter λ is used to avoid the resolution limit. Experiments on synthetic and real-life networks demonstrate the performance and the comparison of experimental results with those of several state-of-the-art methods, the proposed algorithm is considerably efficient and competitive.

1. Introduction

Many complex systems can be represented as graphs or networks. Analyzing the structure of networks or graphs is important for many practical applications in a number of disciplines such as social network analysis, circuit layout problem, image segmentation, and analyzing protein interaction networks, and so on.[1] The related theory has been widely applied in many aspects, including the Internet, communication, biology, and economy. Networks are usually composed of subgroup structures, whose interconnections are dense and intraconnections are sparse. This characteristic in complex networks can be called community structure. Detecting the community structure is one of the fundamental problems for studying networks; it could reveal a latent meaningful structure in networks.[2,3] It is particularly important to detect the structure of commonly used networks, such as daily social networks, recommendation systems, and national power distribution networks.

Community detection is a non-deterministic polynomial problem.[4] The traditional methods of detecting communities in networks are of two categories: graph partitioning and hierarchical clustering. The[5] graph detecting algorithms have been universally applied in information science and other related fields. However, the number of communities and their sizes should to be given before partitioning when using this method.[6] The hierarchical clustering methods include agglomerative clustering algorithm and divisive clustering algorithm, in which the number or size of communities is not required. However, a measure means for specific similarities must be adopted.

Several new methods have been developed to partition and quantify a community structure in complex networks.[7] Newman introduced the modularity as a stopping criterion, which has become one of the most commonly used and best known quality functions. Many modularity-based methods have been proposed, including modularity optimization,[8] simulated annealing,[9] genetic algorithm,[10] extremal optimization,[11] Memetic algorithm, and[12] Spectral optimization. These modularity-based algorithms provide an outstanding way to detect community structures and many studies were carried out using these algorithms.[13] Mahmoud Ahmed and Ibrahem Hafez modeled community problems as a single-objective optimization problem and a multi-objective optimization problem.[14] Lozano and Duch investigated the connection between the dynamics of synchronization and the modularity on complex networks. However,[15] the modularity has the disadvantage of resolution limit as found by Fortunato and Barthelemy. An intrinsic scale and some problems of modularity may not be resolved in an extreme case.[16]

Modularity density as a quantitative measure uses the average modularity degree to evaluate the partition of a network. This approach provides a mesoscopic way of describing the network structure and overcomes the resolution limit of modularity. The partition of community structure will be more accurate if the modularity density is higher. Thus the community partition may be regarded as an optimization issue to discover a partition for a network that has the maximum modularity density.

The estimation of distribution algorithm (EDA) is an intelligent stochastic optimization algorithm.[17] It exploits feasible probabilistic models built around superior solutions (i.e., individuals) while efficiently traversing the search space, which realizes population evolution by the selection procedure, constructing a probability model used to sample next generation individuals. EDA can effectively avoid the specific operation of crossover and mutation in GA, which can essentially solve problems. It is more effective in solving the nonlinear optimization issues and some coupling optimization issues than traditional evolutionary algorithms. However, it is easy to ignore the local information of eminent solutions in the evolution process of EDA. There is no good mechanism to control local optimal solutions.

In this paper, an estimation of distribution algorithm with local sampling strategies (EDALS) is proposed to solve community detection problems. In EDALS, the phenomenon of local community structure was considered by designing a local sampling strategy to improve both the speed and the accuracy for community detection. At the same time, a simulated annealing selection was adopted to enhance population diversity during the evolutionary procedure.

Experimental results on some[18] artificial networks,[19] LFR networks, and[20] authentic networks illustrate the effectiveness of EDALS. Additionally, comparisons of EDALS with other state-of-the-art algorithms demonstrate that the proposed algorithm is competitive.

The remainder of this paper is presented as follows. Some related works are discussed in Sect. 2. A detailed description of the proposed method is given in Sect. 3. In Sect. 4, we demonstrate the experiment on the EDALS in comparison with relative algorithms. In Sect. 5, we summarize the conclusions.

2. Related Works

2.1 *Community structure*

A network can be represented as $G = (V, E)$, where V and E indicate the vertices and links, respectively. Assume that A is the adjacency matrix of G. If there is a link between the nodes i and j, $A_{i,j} = 1$; otherwise, $A_{i,j} = 0$. Suppose that S is a subgraph that belongs to G, and i is a node that belongs to S. k_i is the degree of the vertex i; $k_i^{\text{in}} = \sum_{i,j\in S} A_{i,j}$, $k_i^{\text{out}} = \sum_{i\in,j\notin S} A_{i,j}$.[21] The community of a network usually has the following property:

$$\sum_{i\in S} k_i^{\text{in}} > \sum_{i\in S} k_i^{\text{out}}. \tag{2.1}$$

It means that the interconnections in the community are more than the out-connections toward the rest of the network.

2.2 *Modularity density*

The community structure partition may be constructed as an optimization problem to maximize[7] modularity (Q). Q was proposed by Newman, which is used to discover the community structure of a 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 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. The partition of community structure will be more accurate if the value of Q is higher. Otherwise, the structure is more obscure. The mathematical description of Q is

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{i,j} - \frac{k_i k_j}{2m}\right)\delta(i, j), \tag{2.2}$$

where m is the number of edges in the complex network. If the variables i and j are in the same community, $(i, j) = 1$. Otherwise, $(i, j) = 0$.

A class of approaches used to optimize modularity has been studied. However, modularity optimization has a risk of failing to identify communities smaller than a scale which depends on the total size of the network and on the degree of interconnection density of the modules, even in extreme cases where modules are unambiguously defined.

Modularity density (D) is a new criterion function used to evaluate the community structure of a network, which overcomes the resolution limit in detecting the community structure:

$$D = \sum_{i=1}^{N}\frac{L(V_i, V_i) - L(V_i, \overline{V}_i)}{|V_i|}, \tag{2.3}$$

where $L(V_i, V_i)/|V_i|$ and $L(V_i, \overline{V}_i)/|V_i|$ denote the average internal and external degrees of the i-th 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 strategy to overcome the problem that Q is sensitive to the size of networks and the interconnections of modules. Thus, we could use D to determine whether the networks are partitioned into correct communities. Then[16] a more general version of the criterion function with a tunable parameter λ is introduced to avoid the resolution limit:

$$D_\lambda = \sum_{i=1}^{N}\frac{2\lambda L(V_i, V_i) - 2(1-\lambda)L(V_i, \overline{V}_i)}{|V_i|}. \tag{2.4}$$

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 a small λ; otherwise, small communities are obtained. For this, more details and levels of the network can be found.

2.3 *Local similarity*

The communities can be described as the class of nodes with strong interconnection and spare intraconnections. Thereby, some local similarity metrics should share the same concept of partitioning communities in complex networks. For the positive correlation between the local similarity metrics and the community structures, some typical local similarity metrics such as the[22] Salton similarity measure (Salton),[23] Jaccard similarity measure (Jaccard),[24] Sørensen similarity measure (Sørensen),[25] Revised Sørensen similarity measure (R-Sørensen), and[26] Leicht–Holme–Newman similarity metrics (LHN) were used to represent the local structural description which was associated with community structures in complex networks, and the weight of the links in the complex network needs to be redefined. Salton is represented as the number of common neighbors divided by sqrt($k_i k_j$), where k_i is the degree of the vertex i. R-Sørensen, a revised version of Sørensen, considers the link directly between two nodes.

2.4 *Estimation of distribution algorithms*

Estimation of distribution algorithm is an intelligent evolutional algorithm based on evolution learning and statistical theory, which Substitute probability model and sampling new individuals for crossover and mutation in the genetic algorithm. Many investigations clarify the excellent optimization performance of EDAs to solve either combinatorial optimization or numeric optimization. EDAs mainly contain some algorithms such as the[27] univariate marginal distribution algorithm (UMDA),[28] population based incremental learning (PBIL), compact genetic algorithm (CGA), mutual information maximizing input clustering algorithm (MIMIC), and[29] Bayesian optimization algorithm (BOA).

EDAs address the identification of building blocks using machine-learning techniques to model variables' dependences, from which they can exploit the joint probability of building blocks to generate new solutions. A common EDA framework can be defined using three steps (starting with a randomly generated population):

(1) Select promising solutions
(2) Build a probabilistic model from that selection and
(3) Sample new individuals on the basis of a probability model in a probabilistic approach.

These three steps are repeated until a convergence or stop criterion is met.

3. Description of Proposed Algorithm

Generally speaking, the community structure discovery involves two processes. 1) The network or graph was partitioned into several groups using some detect algorithm. 2) The partition of network was evaluated on the basis of through some criterion. On the basis of evolution learning and statistical theory, a sampling probability model was built according to the excellent individuals selected from the previous-generation population, and then the next-generation evolution population can be generated by sampling the new individuals using the probability model in a probabilistic approach. Thus *EDAs* were improved to solve community detection problems in this paper.

3.1 Framework of EDALS

Considering the application to a large-scale network, an improved *PBIL* was adopted. In the algorithm, the solution set were a binary matrix denoted the community labels and a probability-distributed model was applied to generate promising individuals to obtain excellent performance for solving optimization problems with independent variables in the binary solution space. For a network $G = \{V, E\}$ with node N and the maximum network community label K, the label matrix is defined as $F = [f_{ij}]_{N\times K}, f_{ij} \in \{0, 1\}$ and $\sum_i f_{ij} = 1$. The variable $f_{ij} = 1$ indicates that the community label of node i is j. Thus the probability matrix P is defined as $P = [p_{ij}]_{N\times K}, p_{ij} \in \{0, 1\}$. The variable $p_{ij} = 1$ is the probability that the node i belongs to the community label j and $\sum_i p_{ij} = 1$.

Assuming that the probability matrix of an individual is $P(t) = [p_{ij}(t)]_{N\times K}$ and the number of optimal selection individuals is M, the probability $p_{ij}(t + 1)$ for the label j as the community label of node i in the generation $t + 1$ is calculated:

$$p_{ij}(t+1) = (1-\alpha)p_{ij}(t) + \alpha \frac{\sum\limits_k p_{ij}^k(t)}{M}, \quad 0 < \alpha < 1, \tag{3.1}$$

where the variable $p_{ij}^k(t)$ represents the probability that the label j is selected as the community label of node i in the k-th for M optimal selection individuals, $\sum_k p_{ij}^k(t)$ represents the number that the label j is selected as the community label for node i, and the variable α is the learning rate. The procedure of the proposed method is described as Algorithm 1.

In *EDALS*, the modularity density D described according to Eq. (2.4) as the evaluation function was employed and its maximized value needed to be acquired. In the course of every generation, a truncation selection strategy can be applied to obtain a parent population (Pop), and then a local sampling strategy is carried out to generate new individuals (m_sub_pop). Next, the probability model matrix based on the combination of optima child population (opt_pop-m_pop) and local sampling population (m_sub_pop) is constructed and the final new individual is generated by the genetic sampling operation.

Algorithm 1: Algorithm framework of *EDALS*

```
Input: Gen (Maximum number of generations), Pop_size (Population size), $p_m$ (Mutation probability), Trunc_size=$\xi$*Pop-size, ($0 < \xi < 1$, Truncation selection size), $p_{accept}$ (acception probability), Te (temperature), m_pop_size ← the number of m_pop; $\theta$ ← the ratio of decrease temperature;
initialization: S ← Construct local similarity matrix, Pop ← Generate initial population (Pop_size);
while (Termination condition is not satisfied)
    Fitness evaluation (Pop) according to Eq. (2.4);
    opt_pop ← Truncation selection (Pop, Trunc_size);
    m_pop ← Select m=$\beta$*Trunc_size individals randomly from opt_pop;
    m_sub_pop ← Local sampling operation LSS (m_pop, m_pop_size, Te, $p_m$);
//($0 < p_m < 1$)
    opt_pop′ ← (opt_pop- m_pop) ∪ m_sub_pop;
    $P(t+1)$ ← Updating probability matrix P according to Eq. (3.1);
    Sub_pop ← Sampling operation (P);
    elite ← the fittest individal;
end while
Output: elite.
```

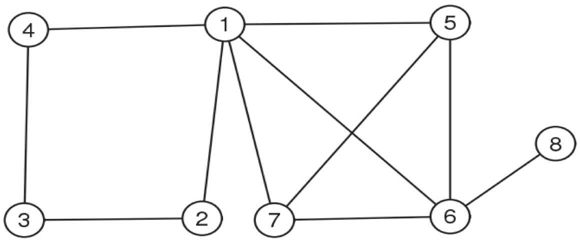

Fig. 1. The local similarity between nodes in the network.

3.2 Local probability matrix

The local probability distribution is significant for implementing the local sampling strategy, which contributes to the effective detection of the community structure by the following proposition. In this paper, there are three steps to acquire the local probability matrix. First is the calculation of the local similarity to obtain a similarity matrix. Second is to obtain the set that the similarity value is not zero for the selected vertex. Finally, the probability distribution matrix can be calculated. Figure 1 shows an example to illustrate the local similarity.

The local probability matrix is constructed on the basis of the local similarity calculated using Eq. (3.2) and the local similarity and probability distribution for each vertex of the network are given in Table I.

$$s_{ij}^{\text{R-Sørensen}} = \frac{e_{ij} + \Gamma(i) \cap \Gamma(j)}{(k_i + k_j)}, \tag{3.2}$$

where $\Gamma(i)$ and k_i indicate the neighbor set and the degree of the vertex i respectively. If the vertices i and j are connected, then $e_{ij} = 1$; otherwise, $e_{ij} = 0$.

The local similarity and probability distribution for each vertex of the network in Fig. 1 are given in Table I.

3.3 Initialization

For each individual, all vertices need to be assigned different numbers standing for different communities in a

Table I. Local similarity and probability distribution for each vertex of the network.

Vertex	Set	Similarity/probability distribution
1	2,3,4,5,6,7,8	0.143, 0.286, 0.143, 0.375, 0.333, 0.375, 0.167 0.078, 0.235, 0.314, 0.520, 0.703, 0.908, 1.000
2	1,3,4,5,6,7	0.143, 0.250, 0.500, 0.200, 0.167, 0.200 0.098, 0.269, 0.612, 0.749, 0.863, 1.000
3	1,2,4	0.286, 0.250, 0.250 0.364, 0.682, 1.000
4	1,2,3,5,6,7	0.143, 0.500, 0.250, 0.200, 0.167, 0.200 0.098, 0.440, 0.612, 0.749, 0.863, 1.000
5	1,2,4,6,7,8	0.375, 0.200, 0.200, 0.429, 0.500, 0.250 0.192, 0.294, 0.397, 0.616, 0.872, 1.000
6	1,2,4,5,7,8	0.333, 0.167, 0.167, 0.429, 0.429, 0.200 0.193, 0.290, 0.387, 0.635, 0.884, 1.000
7	1,2,4,5,6,8	0.375, 0.200, 0.200, 0.500, 0.429, 0.250 0.192, 0.294, 0.397, 0.653, 0.872, 1.000
8	1,5,6,7	0.167, 0.250, 0.200, 0.250 0.192, 0.481, 0.712, 1.000

complex network. For every individual in the first population, the initial number of communities is N. It is a common method to give the *EDAs* not a completely stochastic starting point but a biased one so as to enhance the speed of the algorithm. For every individual, a node is selected randomly and the community number is allotted to its neighboring nodes which mean that the nodes with local similarity to the selected nodes are greater than zero.

Furthermore, a local similarity matrix S according to Eq. (3.2) for a pairwise of vertices in a complex network needs to be constructed, which is regarded as a local sampling probability model used to generate an initialization population.

3.4 *Local sampling strategy*

The local sampling strategy is applied to carry out the local search. Two phases are performed. One is local sampling and the other is simulated annealing selection. It is beneficial to improve either the speed of *EDAs* or the accuracy of community structure detection. The local sampling strategy is presented as Algorithm 2.

Algorithm 2: Algorithm of LSS

```
Input: m_pop, optval, Te, $p_m$.
$T_0 \leftarrow T_0 * \eta$; new_pop ← 0; flag_mute ← 0;
for $m$=1:m_pop_size
    optval ← the fitness of optimal_of_individual;
    for $i$=1:N
        if rand(0,1)<$p_m$
            $\upsilon$(i) ← Find neighbor(i); label(i) ← Find all corresponding label for $\upsilon$(i);
            tmp_individual ← Sampling ($v_i$, label(i), $\upsilon$(i), $S_{i*}$);
            For each r∈label(i)
                curval ← evaluate of current individual;
                $p_{accept}$ ← exp(−(optval − curval)/$T_0$);
                if curval> optval or rand(0,1)<$p_{accept}$
                    optval ← curval;
                    label ← r;
                flag_mute ← 1; new_individual ← tmp_individual;
              end if
          end For
          r(i) ← label;
        end if
    end for
    if flag_mute>0
        new_pop ← new_individual;
    end if
end for
Output: new_pop.
```

In LSS, SA, the probability selection method, will select a substandard individual with some probability, which is favorable for the improvement of the solution in the course of each generation and prevention of a premature in some local optima. Moreover, the process of cooling enables *SA* to obtain the optimum solution gradually. T_0 denotes the initial temperature set to the variance of the individuals in an offspring population. η denotes an annealing ratio range from 0.85 to 0.95. The annealing velocity is inversely proportional to η. p_{accept} is an acceptance probability, whose expression is $p_{\text{accept}} = \exp(-(optval - curval)/T_0)$. The variables *optval* and *curval* are the modularity density of the optimal solution in last generation and the new solution in current generation calculated according to Eq. (2.4) respectively. The probability p_m is a reciprocal of the number of different individuals in a population, which adjusts adaptively with the evaluation of the population. The *FindNeighbors()* function is applied to generate the neighbors of a vertex that is selected randomly in the current individual. At the same time, all corresponding community identifiers are acquired. Then, the community identifier of the selected vertex is sampled on basis of the probability model S_{i^*}.

3.5 *Computational complexity*

A complex network can be represented by a sparse graph. Assuming that n be the number of nodes and e be the number of links in a complex network, the average degree of nodes is $k = e/n$. The lowest convergence speed of the algorithm is the $D_\lambda(i)$ calculation in the local sampling procedure. In the actual operation, there is a duplicate neighbor's label of the selected vertex, which implies that the average number of $D_\lambda(i)$ calculations is not greater than k. Assuming that c be the average size of the community structure, the average time of calculating once should be c owing to the $D_\lambda(i)$ calculation only involving its own community information. Considering that the number of local sampling vertices is n and the number of selected reference vertex for local sampling search is m, the time complexity of LSS is less than $O(mnkc)$. Since m and k are constants, the complexity of the LSS should be $O(cn)$.

Assuming that the evolutionary generation of *EDALS* be L and the population size be M, the lowest convergence speed of the algorithm should be the implementation of the local sampling search. The computational complexity of *EDALS* is not more than $O(\xi LmMcn)$ because of the largest number for executing the local sampling search ξLmM and the complexity of the LSS $O(cn)$. Considering that the parameters M, L, m, and ξ are regarded as constants, the whole computational complexity of *EDALS* is $O(cn)$.

4. Experimental Results

In this section, the proposed algorithm for some artificial networks, LFR networks, and authentic networks will be discussed. The performances of the proposed algorithm were compared with some state-of-the-art algorithms including[30] FN, TGA,[31] Infomap, FTQ, and iMeme-net.

4.1 Experimental settings

In this section, the modularity Q and[32)] normalized mutual information (NMI) were chose to be the measurement criterions when the ground truth of a complex network is known. Otherwise, only the modularity Q is adopted. $NMI(A,B)$ can be calculated as:

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{C_A} C_{i.}\log(C_{i.}/N) + \sum_{j=1}^{C_B} C_{.j}\log(C_{.j}/N)}. \quad (4.1)$$

In this paper, the parameter λ in modularity density D_λ increases from 0.2 to 1.0 at 0.1 intervals. For each value of the parameter λ, all the algorithms are run 30 independent times on the test problems. The number of the population and the maximum number of evaluations were set to 100 and 1.5×10^5 respectively. Among all the results for each network, the best one is selected and shown as the following experiments.

4.2 GN extended benchmark networks

The GN extended benchmark network contains four communities with 128 vertices, and each community has 32 nodes. The average degree of node is 16, and the mixing parameter μ determines the percentage of connections between communities to the total connections. When the express $\mu < 0.5$ is true, the network has a clear group structure. On the other hand, when the express $\mu > 0.5$ is true, the community structure is vague, and it is difficult to detect its structure.

Experiments on GN extended networks were carried out to test the performance of our algorithm. FN, TGA, Infomap, FTQ, and iMeme-net and the proposed algorithms were tested on 10 GN extended networks with the mixing parameter μ distributing from 0.05 to 0.5. Figure 2 shows the average maximum NMI and μ values obtained by different methods when the parameter μ increasing from 0.05 to 0.5 at 0.05 intervals.

As shown in Fig. 2, the superiority of the proposed algorithm EDALS is demonstrated. EDALS shows its excellent detection capability when the community structure becomes increasingly obscure with the change in the parameter μ. The algorithm iMeme-net cannot detect the real community structure in any mixing parameter μ. The algorithms FN, TGA, Infomap, FTQ, and the proposed algorithm can acquire the true community structure when the parameter $\mu \leq 0.15$. With the increase in the parameter μ, the community structure of the complex network becomes fuzzy and it becomes difficult to detect the true structure of the community. Informap and FN first show their weakness. The detection capability of Informap decreases rapidly from $\mu = 0.15$ to 0.3 and the performance of FN decreases gradually from $\mu = 0.1$ to 0.5. When $\mu > 0.3$, the algorithms TGA and FTQ show their limitation in detecting the community structure. By comparing these results for selected algorithms and EDALS performed on GN extended benchmark networks, the algorithm EDALS shows its superiority. From our view point, the designed definition takes the topology of the community structure into consideration, which allows the algorithm to detect a more obscure structure than the others with a suitable tuning parameter value in the objective functions.

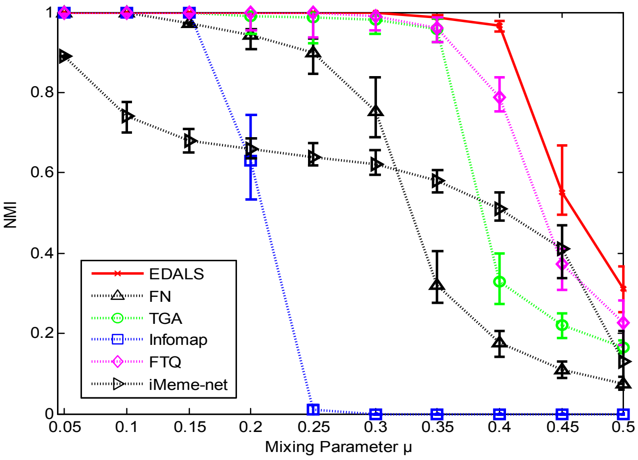

Fig. 2. (Color online) Average maximum NMI over 30 runs on GN extended benchmark networks.

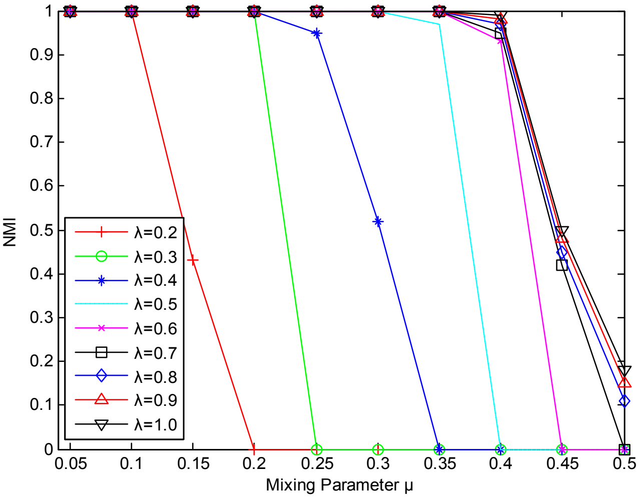

Fig. 3. (Color online) Average maximum NMI with EDALS over 30 runs on GN extended benchmark networks.

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

Figure 3 shows the simulation results executed by the algorithm EDALS over 30 runs under different mixing parameters μ on GN extended benchmark networks. As seen from Fig. 3, the algorithm EDALS can be used to detect the community structure under the condition that the value of the parameter μ is less than 0.35. When the parameter λ is greater than 0.7, the algorithm EDALS can obtain good results.

To discuss the convergence of the nature-inspired algorithms (the proposed algorithm, TGA, and iMeme-net algorithm), the GN extended benchmark network $\mu = 0.3$ was chosen to illustrate the convergence procedure.

As shown in Fig. 4, NMI converges to 1 or almost 1 in about 4×10^4 evaluations. The exact number of communities cannot be detected by the algorithm iMeme-net because the NMI values are convergent to 0.624. TGA can obtain a satisfying result, but it has a slow convergence speed. Thus a conclusion can be derived that EDALS significantly outper-

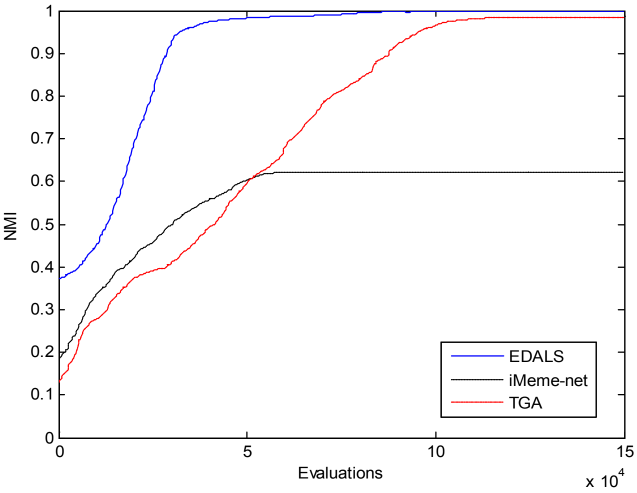

Fig. 4. (Color online) Convergence of three algorithms.

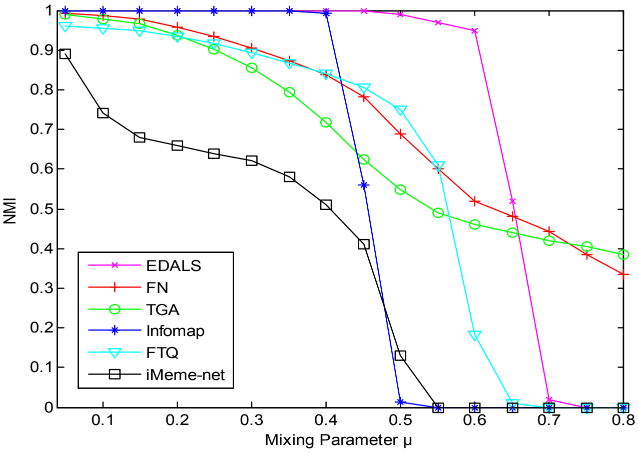

Fig. 5. (Color online) Average maximum NMI on LFR networks.

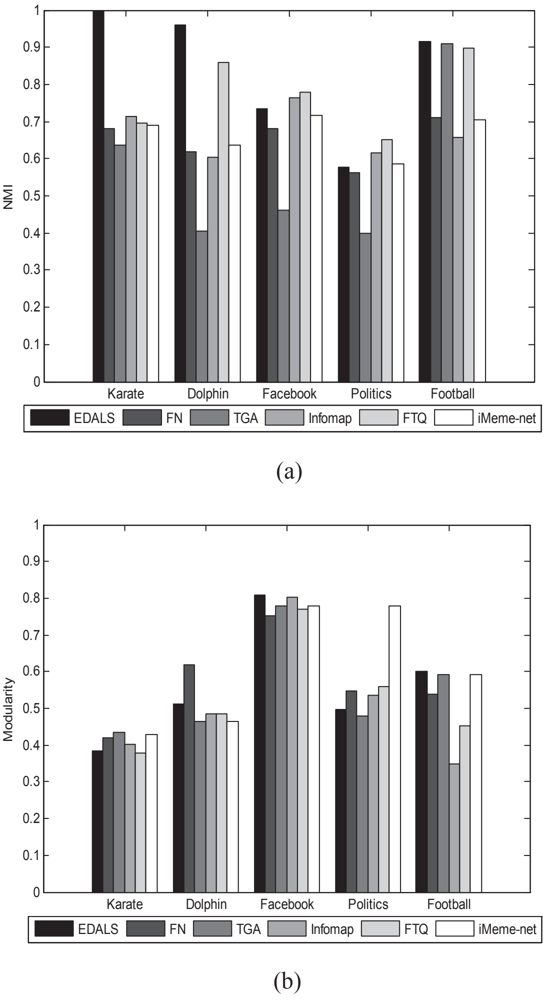

Fig. 6. NMI and modularity for each dataset determined by each algorithm.

forms other relative algorithms and is competent to detect the real structure of a network effectively from Fig. 4.

4.3 *LFR benchmark network*

In LFR benchmark networks, the parameters γ and β are set to tune the distribution of the degree and community size. Each vertex shares a fraction $1-\mu$ of its edges with the other vertices in the same community and the fraction μ with the vertices in other communities.

In this model, the mixing parameter changes in $[0.05, 0.8]$ at 0.05 intervals. Each of them consists of 1000 vertices. The averaged degree for each node is 20 and the maximum node degree is 50.

As shown in Fig. 5, EDALS has a better performance than the other five methods on LFR networks. It can be concluded that the initialization and local sampling strategy of EDALS can reveal the community structure more accurately than the other five algorithms as shown by the comparison result. The performance of iMeme-net is the worst among all the six algorithms because it uses a different encoding scheme.

4.4 *Real-life networks*

To further test the effectiveness of the proposed algorithm, the experiments were performed on five well-known real-life networks, namely, the[33] Zacharys Karate Club network, Dolphin social network, American College Football network,[34] Krebs Books on US Politics network, and Facebook network.

Comparison experiments of NMI and modularity on the five real-life networks were carried out and the results are shown in Figs. 6(a) and 6(b), respectively. The algorithm EDALS demonstrates a better performance than the relative given algorithms from what is recorded on the Zacharys Karate Club, Dolphin, Facebook, Krebs Books on US Politics, and American College Football networks. In the case of the Krebs Books on US Politics network and the Facebook network, the performance of the proposed algorithm EDALS is slightly worse than those of the FTQ and Informap algorithms.

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 Fig. 7. It is clear that the proposed algorithm has a comparative time cost, and TGA is the most time-consuming since the time complexity of the decoding step is greater than those of the others.

From the above experiments, EDALS is considerable more efficient and competitive in solving community structure partition problems than FN, TGA, Infomap, FTQ, and iMeme-net. The local sampling strategy contributes to the

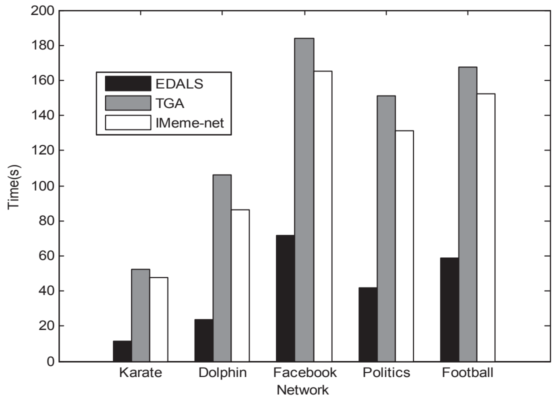

Fig. 7. Computational time of three algorithms.

improvement of executive speed for the EDAs and the accuracy of community detection, and simulated annealing selection can increase population diversity in the evolutionary procedure.

5. Conclusions

In this paper, an estimation of the distribution algorithm with local sampling strategies is proposed to solve community detection problems using modularity density as a fitness function. The sampling probability matrix can be acquired from the local similarity matrix, by which the local sampling strategy is developed to improve convergence speed and accuracy. Furthermore, a biased initialization of population is designed using a local similarity matrix to speed up the convergence. To avoid EDALS stall at local optimums, a simulated annealing selection is adopted. The performance of EDALS was tested on some artificial networks, LFR networks and several authentic networks with known community structures. The experiments on them showed that the performance of EDALS was good and modularity density maximization can resolve the resolution limit problem. In the future, we can improve our algorithm to reveal communities on large-scale networks.

Acknowledgments

This research was supported by the Zhejiang Provincial Education Department Research Foundation of China under Grant No. Y201533771, Zhejiang Provincial Natural Science Foundation of China under Grant Nos. LY16F020027 and LY15F020040, and Humanity and Social Science Youth Foundation of the Ministry of Education of China under Grant No. 15YJCZH088.

*fhyu520@whu.edu.cn

1) A. J. Alejandro, C. E. Sanz-Rodríguez, and J. L. Cabrera, Philos. Trans. R. Soc. Am. **373**, 20150108 (2015).
2) B. Ball, B. Karrer, and M. E. J. Newman, Phys. Rev. E **84**, 036103 (2011).
3) Z. Li, S. Zhang, and X. Zhang, Am. J. Oper. Res. **5**, 421 (2015).
4) S. Gómez, P. Jensen, and A. Arenas, Phys. Rev. E **80**, 016114 (2009).
5) N. Aston and W. Hu, Commun. Network **6**, 124 (2014).
6) S. Fortunato, Phys. Rep. **486**, 75 (2010).
7) M. E. J. Newman, Eur. Phys. J. B **38**, 321 (2004).
8) J. Liu and T. Liu, Physica A **389**, 2300 (2010).
9) C. Pizzuti, in *Parallel Problem Solving from Nature — PPSN X*, ed. G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni (Springer, Heidelberg, 2008) Lecture Notes in Computer Science, Vol. 5199, Chap. 107.
10) J. Duch and A. Arenas, Phys. Rev. E **72**, 027104 (2005).
11) M. Gong, Q. Cai, Y. Li, and J. Ma, IEEE Congr. Evolutionary Computation, 2012, Vol. 18, Suppl. B, p. 1.
12) Z. Wang, Z. Chen, Y. Zhao, and S. Chen, Sci. World J. **2014**, 329325 (2014).
13) M. M. Ahmed, A. I. Hafez, M. M. Elwakil, A. E. Hassanien, and E. Hassanien, in *The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28–30, 2015, Beni Suef, Egypt* (Springer International Publishing, Cham, 2016) Advances in Intelligent Systems and Computing, Vol. 407, Chap. 12.
14) S. Lozano, J. Duch, and A. Arenas, Eur. Phys. J. D **143**, 257 (2007).
15) S. Fortunato and M. Barthelemy, Proc. Natl. Acad. Sci. U.S.A. **104**, 36 (2007).
16) Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, Phys. Rev. E **77**, 036109 (2008).
17) M. Hauschild and M. Pelikan, Swarm Evol. Comput. **1**, 111 (2011).
18) A. I. Hafez, A. E. Hassanien, and A. A. Fahmy, Soc. Networks **65**, 85 (2014).
19) A. Lancichinetti, S. Fortunato, and F. Radicchi, Phys. Rev. E **78**, 046110 (2008).
20) Q. Cai, L. Ma, M. Gong, and D. Tian, Int. J. Bio-Inspired Comput. **8**, 84 (2016).
21) J. Huang, Y. Sun, Y. Liu, and B. Wang, Int. J. u- e- Serv. Sci. Technol. **8**, 51 (2015).
22) S. Li, Y. Chen, H. Du, and M. W. Feldman, Complexity **15**, 53 (2010).
23) T. Heimo, J. M. Kumpula, K. Kaski, and J. Saramäki, J. Stat. Mech. **2008**, P08007 (2008).
24) Y. Zhou, G. Sun, Y. Xing, R. Zhou, and Z. Wang, Appl. Comput. Intell. Soft Comput. **2016**, 3217612 (2016).
25) H. Okamoto, Biosystems **146**, 85 (2016).
26) E. A. Leicht, P. Holme, and M. E. J. Newman, Phys. Rev. E **73**, 026120 (2006).
27) A. R. Gonçalves and F. J. Von Zuben, Proc. IEEE Congr. Evolutionary Computation **2011**, 62 (2011).
28) C. Fyfe, Soft Comput. **2**, 191 (1999).
29) M. Pelikan, D. E. Godberg, and E. C. Paz, Evol. Comput. **8**, 311 (2000).
30) M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).
31) M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall, Phys. Rev. X **5**, 011027 (2015).
32) C. Liu, J. Liu, and Z. Jiang, IEEE Trans. Cybern. **44**, 2274 (2014).
33) D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behav. Ecol. Sociobiol. **54**, 396 (2003).
34) F. Wu and B. A. Huberman, Eur. Phys. J. B **38**, 331 (2004).