2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

An Intelligent Ant Colony Optimization for Community Detection in Complex Networks

Caihong Mu, Jian Zhang, Licheng Jiao
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education
International Research Center for Intelligent Perception and Computation
Xidian University, Xi'an, China
E-mail: mucaihongxd@foxmail.com

***Abstract*—Many systems in social world can be represented by complex networks. It is of great significance to detect the community structure and analyze the functions for networks. In recent years, plenty of research and works have been focused on this problem. In this paper, we propose an enhanced algorithm based on ant colony optimization (ACO) for the community detection problems. In order to avoid redundant computing in ACO, we divide the ant colony into two groups, original group and intelligent group, which search the solution space simultaneously. In the intelligent group, due to the locus-based adjacency representation of the solution, we let some of them have an ability of self-learning and others can learn from the optimal solutions proactively. Experiments on synthetic and real-life networks show the proposed algorithm can explore in an efficient and stable way.**

Keywords—complex networks; community detection; intelligent ant clony optimization; self-learning; proactive-learning

I. INRODUCTION

The form of complex networks can be commonly discovered in the world. As been called graphs mathematic, networks involve many different areas such as sociology, biology, mathematics, and physics. In the last decade, many works had been focused on the research of the components and functions of a specific network by applying graph theory [1]. When we consider the community detection problems as a graph analysis, a node or vertex equals to a dot and an edge connecting two vertices equals to a line [2].

In the past decade, many methods were proposed to identify modules and classify vertices according to the principle that the connections should be sparse between communities and be dense inside [3][4]. Therefore, some objective functions which measure the quality and veracity of a partition have been gradually proposed. For example, the *modularity* introduced by Girvan and Newman in [5], which is widely accepted and most commonly used in many algorithms. However, the method of optimizing the modularity Q has a resolution limitation problem according to the theory presented by Santo Fortunato and Marc Barthélemy [6]. Then a new quality function was introduced by Li et al. to overcome the limitation of modularity called modularity density [7]. And they also theoretically proved out that the modularity density can reveal communities of multi-resolution with a tunable parameter λ as a general version. Gong et al. [8] took the modularity density apart into two components ratio association [9] and ratio cut [10]. And due to the frame of multiobjective evolutionary algorithm, they can receive various different partitions hierarchically.

Among plenty of techniques for community detection problem, the family of modularity based optimization algorithms play an important role. Actually, Modularity is originally proposed as a stopping criterion in the algorithm of Girvan and Newman (GN). The first algorithm to maximize modularity called FN in a greedy way by Newman [11]. It is a clustering method which can uncover the hierarchical structure by merging vertices with a maximum growth of value Q. Guimerà et al. [12] took use of simulated annealing to optimize modularity which result in reducing the risk of getting trapped in local optimum. However, the performance of simulated annealing is sensitive to the parameters (initial temperature and cooling factor). Duch and Arenas employed the extremal optimization which is a heuristic search strategy for the first attempting to maximizing modularity [13]. Newman found a spectral algorithm which can express the modularity in terms of eigenspectrum of a matrix [14][15]. And this procedure also aims at maximizing modularity.

In this paper, we propose a modified algorithm called intelligent ant colony optimization for networks (IACO-Net) based on the optimization of modularity. In order to avoid redundant computing during the convergence process of ACO, We divide the ant colony into two groups, original group and intelligent group, which search the solution space simultaneously. For the intelligent ants in intelligent group, we assume that some of them have an ability of self-learning and others can learn from the optimal solutions proactively. And the locus-based encoding method is applied in the representation of solutions. By comparing with ACO and other algorithms, the proposed IACO-Net converges faster and receives a higher modularity.

The rest of this article is structured as follows: section two introduces some related works about ant-based optimization and represents the definition of community detection. Section three shows the procedure of the proposed algorithm in detail. And the results of our experiments are represented in the section four. The last section includes the conclusions and the works in the future.

This work was supported by the National Basic Research Program (973 Program) of China (No. 2013CB329402), the National Natural Science Foundation of China (Nos. 61003199, 61072139, 61373111，61272279, 61303032 and 61001202) and the Fundamental Research Funds for the Central Universities (Nos. JB140216, K5051202019, K5051302049 and K50511020014).

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

II. Related Work

A. Problem Definition

As we have mentioned above, a complex network can be regarded as a graph $G = (V, E)$, which includes $N = |V|$ vertices and $m = |E|$ edges. V represents the vertices set $\{v_1, v_2, \dots, v_N\}$, and E represents the set of edges in G. A is an $N \times N$ adjacent matrix of an unweighted graph G, the element A_{ij} is 1 when there is an edge between vertex v_i and vertex v_j, otherwise A_{ij} equals 0. For a final partition $P = \{C_1, C_2, \dots, C_K\}$ of a network, each element C_l $(l = 1,2,\dots,K)$ is a proper subset of V, and K is the total number of communities.

For a particular partition of a network, how to assess its quality should be emphasized on. A significative quality metric called modularity Q has been introduced by Girvan and Newman [4], which is shown as follows:

$$Q = \frac{1}{2m}\sum_{ij}\left(\left(A_{ij} - \frac{k_i k_j}{2m}\right) \times \delta\big(C(i), C(i)\big)\right) \qquad (1)$$

Where m is the number of edges in the network and $m = \frac{1}{2}\sum_{ij} A_{ij}$, A_{ij} is an element of the adjacent matrix A, $k_i = \sum_j A_{ij}$ is the degree of vertex i which equals to the total number of neighbors of vertex i. $\delta(u, v)$ is the Kronecker delta function in which $\delta(u, v) = 1$ if $u = v$, otherwise $\delta(u, v) = 0$. And $C(i)$ is the index of the community that vertex i belongs to. If the partition shows the number of edges within community is less than a random network, this quantity will get $Q = 0$. Actually, value Q of most social networks fall in the range about 0.3 to 0.7 [4], and high values of modularity indicate good partitions [1].

B. Ant-based Algorithms for Community Detection

Inspired by the behavior of social insects, technologists produced a series of new solutions for traditional problems through a simulation on social insects. These works were called swarm intelligence which refers to subjects without intelligence exhibit intelligent behavior through cooperation. And the swarm are a group of subjects which can communicate with each other directly and indirectly. Under these conditions, swarm intelligence provides the basis for solving complex distributed problems without centralized control and global model. There are two kinds of algorithms based on swarm intelligence in the field of computational intelligence, Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO).

In this article, we focus on the ACO algorithm proposed by Marco Dorigo [16] which is a probability-based technology inspired by the behavior of ant colonies in the process of foraging. Actually, what an ant concerns is not the information of the global world but the front information in a small area. Then they can utilize several simple rules to make decisions according to the local information. Thus, complex behavior will be highlighted in the swarm of ant colonies.

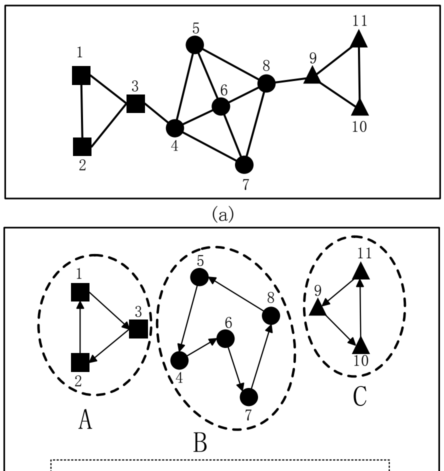
Due to the diversity and positive feedback, two important properties of the ant-based algorithm, ant colonies have an outstanding performance in solving many problems, such as traveling salesman problem (TSP) [17], job shop scheduling problem [18], graph coloring [19] and vehicle routing problem [20]. It is the diversity that guarantee the ants not trapping in a blind alley in the procedure of foraging. And it is the positive feedback mechanism ensure the better solution to be preserved. Definitely, a tradeoff decision between these two properties will be determined finally.

In the field of community detection, ant-based algorithms also have been attempted to find groups in complex networks and experimentally verified to be feasible and efficient. Ji. et al. [21] proposed an ant colony clustering algorithm (called ACC-FP in this paper) in which the clustering process occurs in a virtual grid. And a pheromone diffusion model is employed by the pheromone trail update strategy which depends on the distribution in the grid and the structure of network simultaneously. Despite undirected social networks, community detection in oriented social networks also attracts attention in [22], the focus of the method put forward in which is the employing of ant colony optimization. Based on the locus-based coding strategy, Chang. et al. [23] introduced an ACO approach to find the structure of social networks. The proposed algorithm is under the framework of Max-Min Ant System (MMAS) [24] and has some features to adapt to the characteristics of community detection problem.

III. The Iaco-net Algorithm

A. Representation

In this algorithm, solutions are coded by the locus-based adjacency representation [25]. The most significant advantage of this representation is the automatical decoding process without knowing the clusters number in advance. For a particular network with N vertices, every individual g in the population is expressed as $\langle g_1, g_2, \dots, g_N\rangle$ with N genes. And each g_i can take the allele value j in the range of $\langle 1,2,\dots,N\rangle$. Fig. 1(a) shows a hypothetical network contains 11 vertices in three different shapes of square, circle and triangle. And a solution of locus-based expression is shown in Fig. 1(b). Obviously, the network consists of three components, community A includes 3 vertices $\{1,2,3\}$, community B includes 5 vertices $\{4,5,6,7,8\}$ and the rest vertices $\{9,10,11\}$ belong to community C. The community label of each vertex


1
2
3
4
5
6
7
8
9
10
11
A
B
C
Position: 1 2 3 4 5 6 7 8 9 10 11
The Ant: 3 1 2 6 4 7 8 5 10 11 9

(a)

(b)

Fig. 1. (a) A model of a network with 11 vertices and 16 edges, including 3 communities expressed in different shape. (b) The locus-based adjacency scheme, the "Ant" sequence and the corresponding partition (including 3 communities: A, B and C) is is presented in Fig. 1(b) intuitively.

B. Solution Construction

Due to the locus-based adjacency representation in previous, a solution can be constructed by a corresponding ant. Each ant takes charge of choosing the allele values for all the N vertices by a probabilistic choice model of the ACO algorithm. In other words, a particular ant selects N edges in turn which may include two same edges. The probability $P_{ij}(t)$ of current vertex i choosing the next vertex j involve two factors, the pheromone information τ_{ij} and the heuristic information η_{ij}. Both of them are of great importance to determine the probability ratio among the neighbors. And the probabilistic model for vertex i to choose a connected vertex j is introduced as follows:

$$P_{ij}(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in \mathcal{N}(i)} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}}, & j \in \mathcal{N}(i) \\ 0, & otherwise \end{cases} \tag{2}$$

$$\eta_{ij} = \frac{1}{1 + e^{-C(i,j)}} \tag{3}$$

$$C(i,j) = \frac{\sum_{v_k \in V} (A_{ik} - \mu_i)(A_{jk} - \mu_j)}{n\sigma_i\sigma_j} \tag{4}$$

Where $\mathcal{N}(i)$is the neighbor vertices set of vertex i in the network. The fraction consists of two components, the pheromone trail information $\tau_{ij}(t)$ and the heuristic information η_{ij}. The former is the value of pheromone trail on the edge $\langle i,j \rangle$, and the latter is the structure similarity between vertex i and vertex j in the $G = (V, E)$. The parameters α and β determine the importance of pheromone trail factor versus heuristic factor. Obviously, the higher the value of $\tau_{ij}(t)$ and η_{ij}, the more possible for source vertex i to select vertex j.

In (4), $C(i,j)$ is the Pearson correlation between vertices v_i and v_j, A_{jk} is the corresponding element in the adjacency matrix$A$, $\mu_i = \sum_k A_{ik}/N$ and $\sigma_i = \sqrt{\sum_k (A_{ik} - \mu_i)^2/N}$. And a higher value of η_{ij} denotes a higher similarity between vertex i and vertex j.

In addition, there is a pseudo-random probabilistic model including a threshold value q_0 which is smaller than 1. Before computing the $P_{ij}(t)$, if a random number q is larger than the threshold q_0 , the allele vertex j will be determined according to the $P_{ij}(t)$ by the roulette selection mechanism. On the contrary, the choice of vertex j will be the vertex with the maximal value of $\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}$. When $q \leq q_0$, vertex j can be determined by the following rule:

$$j = arg\, max_{j \in \mathcal{N}(i)} \left\{ \left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta} \right\} \tag{5}$$

In our algorithm, in order to compare with the ACO algorithm in [23], we follow the original probabilistic model according to (2).

After getting a solution consists of N edges by an ant with N steps, a decoding procedure will be adopted to transform the locus-based solution into a certain community label set. Thus, modularity Q will be calculated according to (1). This procedure will be repeated N_a times during each iteration because the amount of the ant colony population is N_a.

C. Proactive-learning Mechanism

As our observation, as well as other problems using ACO [25]-[27], the pheromone factor τ_{ij} of some edges has a trend to be closer to each other as the iterations increase. After several iterations, the pheromone on such edges with a higher η_{ij} (which means vertex i and vertex j are more likely belong to the same community) will be relatively more than that of other edges. More precisely, the ants will favor these edges called sub-solutions, which may eventually be parts of the final solution. Obviously, in a particular iteration, a great many ants choose the same edges, this phenomenon will lead to some redundant computations, such as repeatedly computing of τ_{ij} and P_{ij} for the current vertex i. As the ACO converges, this situation will become more and more serious. Actually, this situation also happened in other metaheuristics, such as GA-Net [28].

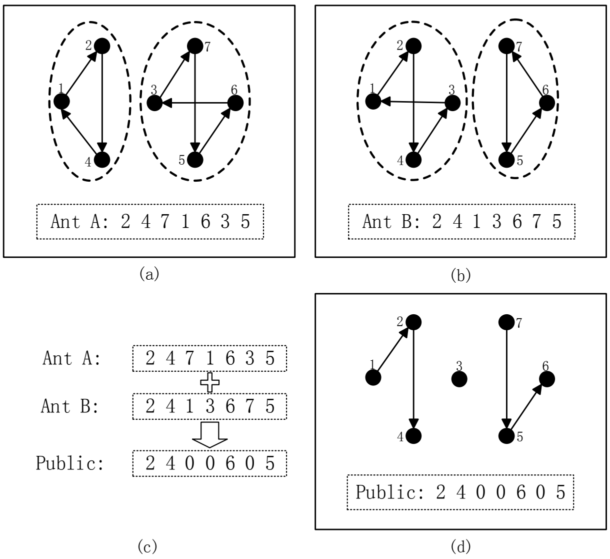


Fig. 2. (a) and (b) are two locus-based solutions, "Ant A" is the optimal solution and "Ant B" is the sub-optimal solution. (c) and (d) introduce the generation of public edge in detail.

Fig. 2 shows a schematic to illustrate the repeated edges that may lead to redundant computations during the convergence process of ACO. Fig. 2(a) and Fig. 2(b) show two possible solutions for a hypothetical network of seven vertices. And we assume the two partitions are the optimal and sub-optimal solutions which we found in present. So we can intuitively extract the public part $\langle 2,4,0,0,6,0,5 \rangle$ of the two ants A: $\langle 2,4,7,1,6,3,5 \rangle$ and : $\langle 2,4,1,3,6,7,5 \rangle$, and the public part actually includes four public edges $\langle 1,2 \rangle$, $\langle 2,4 \rangle$, $\langle 5,6 \rangle$ and $\langle 7,5 \rangle$ which means that these edges are more likely to be components of the global optimal solution. This characteristic of ACO for community detection stimulates our interest in how to make use of the public edges to avoid redundant computing and reduce the computation time.

In order to improve this phenomenon, we propose a modified algorithm called intelligent ACO. Our motivation is to improve the searching speed and the efficiency to find global optimum during the optimization searching process on the basis of public edges (PE for short). We separate a small part of ants from the colony, and these separated ants have an intelligent ability of learning from the optimal and sub-optimal solutions. In each iteration, after all the N_a ants have been constructed, the optimal and sub-optimal solutions will be saved according to the value of modularity. Then compare the optimal and sub-optimal solutions of current iteration and of last iteration, and the two ants with higher value of modularity among the four individuals will be the "Ant A" and "Ant B" in the Fig. 2 to update the public edges (despite for the first iteration, which the public edges are initialized with a null vector).

Assuming the locus-based public edges represented as a vector $PE: (p_1, p_2, \dots, p_N)$, of course some of the elements are 0. The amount of the intelligent ants is N_{Ia}. Each intelligent ant construct a solution on the basis of PE.

For a current vertex i, if $p_i \neq 0$, there will be no need to evaluate the probabilistic model, and directly let $j = p_i$, otherwise, if p_i in PE equals to 0, the choice of edge $\langle i, j\rangle$ is still determined by P_{ij} in a roulette way. Therefore, the probabilistic model of intelligent ant $P_{ij}^{Ia}(t)$ can be defined as:

$$P_{ij}^{Ia}(t) = \begin{cases} 1(j = p_i) & v_i \neq 0 \\ P_{ij}, & v_i = 0 \end{cases} \qquad (6)$$

D. Self-learning Mechanism

In this part, we assume another kind of intelligent ant has an ability of self-learning. It is a mechanism that searching for a better solution by itself without the knowledge of the prior information and global optimal information. In order to reduce the computational complexity, we just allow the optimal ant and sub-optimal ant in the whole ant colony to have such an ability. Then each ant randomly pick $\lceil \lambda \cdot$ *Na* vertices out and replace each vertex with its connected vertices orderly, then evaluate the value Q correspondingly. Where λ is a tunable parameter smaller than 1 and $\lceil x \rceil$ denotes rounding the elements of x to the nearest integers towards infinity. Finally, if a better Q has been found, the original ant will be replaced by the new ant with the biggest Q. And the procedure can be described as pseudo-code in *Algorithm 1.*

Thus, benefit from the relatively better characteristics and structure, these two clever ants are more likely to find a much better solution which is closer to the global optimum.

Algorithm 1: Framework of *self-learning ().*

```
Input: a solution $S = \{v_1, v_2, \dots, v_N\}$
Parameters: λ
  Choose $\lceil \lambda \cdot N \rceil$ locations in $S$ randomly
  for (each randomly location $i$ in $S$ )
    for (each neighbor $j$ of $v_i$)
      $s' \leftarrow v_i$ replaced by neighbor($j$);
      if (modularity($s'$) > modularity($s$))
        $s \leftarrow s'$;
    end for
  end for
Output: a solution with the higher modularity
```

E. Pheromone Trail Update

The most virtual and important differences between all the ACO-based algorithms are pheromone trail definition and update mechanism. The pheromone trail update mechanism in this paper follows the MMAS.

The MMAS limits the pheromone trails to an interval $[\tau_{min}, \tau_{max}]$ to avoid early search stagnation. The pheromone trail on each edge should be deliberately initialized to τ_{max} with a purpose of achieving a higher exploration of solutions at the start of the algorithm. And to exploit the best solutions found during the current iteration, only one ant adds pheromone after each iteration. This ant in our algorithm is the best solution of current iteration and only the edges in the iteration-best solution have their pheromone trail increasing. Therefore, the update mechanism of the pheromone trail can be represented as follows:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \qquad (7)$$

Where the parameter ρ (with $0 \leq \rho < 1$) in the trail persistence (then, $1 - \rho$ models the evaporation) and $\Delta\tau_{ij}$ denotes the amount of pheromone increased on the edge $\langle i, j\rangle$, and it can be written as follows:

$$\Delta\tau_{ij} = \begin{cases} Q_{ib} & edge\ \langle i, j\rangle \in S_{ib} \\ 0 & otherwise \end{cases} \qquad (8)$$

Where S_{ib} is the best solution of the current iteration, Q_{ib} is the modularity Q of the solution correspondingly.

Due to the pheromone limitation suggested by MMAS, we limit the pheromone amount on each edge to a range of an interval $[\tau_{min}, \tau_{max}]$. And the maximum pheromone trail τ_{max} is determined by an estimate of the asymptotically maximum value. The τ_{max} is defined as follows:

$$\tau_{max} = \frac{1}{1-\rho} \cdot Q_{gb} \qquad (9)$$

Where Q_{gb} is modularity of global optimal solution with the largest Q found so far.

The definition of τ_{min} in the MMAS is:

$$\tau_{min} = \frac{\tau_{max}(1 - \sqrt[n]{p_{best}})}{(\frac{N}{2} - 1)\sqrt[n]{p_{best}}} \qquad (10)$$

Where p_{best} is the probability to find the best solution. However, in order to simplify the expression of the equation, we redefine the τ_{min} as follows:

$$\tau_{min} = \varepsilon \cdot \tau_{max} \qquad (11)$$

Where the coefficient ε is a fairly small number.

During the procedure of exploring, when a new global optimal solution was found, τ_{max} and τ_{min} would be updated according to (9) and (11). The complete pseudo-code of IACO-Net algorithm for community detection in complex network is shown as *Algorithm 2.*

In this way, the update of pheromone trail consists of two stages: the weakening stage and reinforcement stage. During the pheromone weakening stage, as a global updating, the pheromone trails on all the edges are reduced by the evaporation factor. This procedure can avoids early search stagnation due to the excessive depositing of pheromone, and encourage the ants in the next iteration to search for new solution. During the reinforcement stage, a positive feedback of pheromone strategy is adopted on the edges of the iteration-solution merely. This stage makes sure the prior information of good solutions can be utilized and promotes the convergence rate of the algorithm to find optimal solution.

***Algorithm 2*: IACO-Net.**

```
 Input: A complex network modeled by $G = (V, E)$
Output: $P = (C_1, C_2, \dots, C_k)$, the partition of the network
1.Initialization:
    Set parameters $t, N_a, \tau(0), \alpha, \beta, \rho, \lambda$;
    Initialize the public edge $PE = zeros(1, V)$;
    Evaluate $\eta$ of each edge $\langle i, j\rangle$ in graph G;
    Divide the colony into two groups: intelligent ants($N_{Ia} = \lambda \cdot N_a$) and original ants($N_{Oa} = N_a - N_{Ia}$).
2.Intelligent ACO stage:
    while ($t < Max\ Number\ of\ Generations$) do
        for $i = 1\ to\ N_{Oa}$ do
            {Construct a new solution according to (2)};
        for $j = 1\ to\ N_{Ia}$ do
            //proactive-learning process.
            {Construct a new solution according to (2)};
        Self-learning(the optimal solution)
        Self-learning(the sub-optimal solution)
        Update the iteration-best ($S_{ib}$) and global-best ($S_{gb}$)
        Compute pheromone trail limits $\tau_{max}$ and $\tau_{min}$
        Update public edges ($PE$)
        Update pheromone trails
    end while
3.Output the partition with maximum modularity
```

IV. Empirical Study

In order to verify the effectiveness of our method, we apply the IACO-Net algorithm to a synthetic network and four real-world networks. In this section, the simulation experiments will be showed, and the analysis of the results are presented in detail. What is worth mentioning is the algorithms that we adopted in the contrast experiment. The first one is the algorithm FN which can reveal hierarchical structure of the network, and this greed technique to optimize modularity Q was proposed by Newman [29]. The FN algorithm also has an advantage of running speed. The other one is GA-Net, a genetic-based method proposed by Pizzuti [28]. The GA-Net can identify communities in the network by optimizing a simple effective fitness function *community score*.

We simulate the proposed algorithm and two compared algorithms in MATLAB. All the experiments have been performed on a PC with an Intel(R) Core(TM)2 Duo CPU, 2.20 GHz, 2 GB RAM under Windows 7 OS. Specially, some of the parameters in IACO-Net follow the ACO that are set to typical values constantly. For instance, we set $\alpha = 1, \beta = 2, \rho = 0.8, \varepsilon = 0.005$. In addition, the number of the whole ant colony $N_a = 50$ including intelligent ants of $N_{Ia} = 20$ and original ants of $N_{Oa} = 30$. The coefficient in the self-learning procedure $\lambda = 0.2$. And the termination condition use a maximum number of iteration which is set to 100.

A. Evaluation Criteria

For the modularity-optimization based algorithm, another evaluation criteria is definitely needed to measure the similarity between the structure revealed by the proposed algorithm and the real structure. In consequence, the *normalized mutual information* (NMI) in [30] stands out due to its reliability and veracity.

By assuming that there is a network with two partitions A and B, we let C be a confusion matrix whose element C_{ij} is the number of the same vertices in the community i of partition A and in the community j of partition B. Then the NMI is defined as:

$$\mathrm{NMI} = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log\left({C_{ij}N}\Big/{C_{i.}C_{.j}} \right)}{\sum_{i=1}^{C_A}\left(C_{i.}\log\left({C_{i.}}/{N}\right)\right)+\sum_{j=1}^{C_B}\left(C_{.j}\log\left({C_{.j}}/{N}\right)\right)} \quad (12)$$

Where $C_A(C_B)$ is the number of communities in the partition $A(B)$, $C_{i.}(C_{.j})$ is the sum of the elements in row i (column j), and N is the number of vertices. Obviously, $I(A, B) = I(B, A)$, if $A = B$, $I = 1$, and $I = 0$ when A is totally different with B.

Generally, a higher value of NMI corresponds to a closer partition to the real one. What should be noticed is value Q and value NMI are two different metrics, there does not exist a positive correlation between them. In some cases, a partition with a higher Q may not share a higher NMI. This phenomenon will be met in the following part.

B. Results on Synthetic Networks

Firstly, we apply our algorithm to the benchmark networks (LFR) proposed by Lancichinetti et al. in [31]. The LFR benchmark is an extension of the classical benchmark in [32] proposed Grivan and Newman. And the LFR benchmark contains 128 vertices with 4 clusters, and each cluster including 32 vertices. Each vertex has an average degree of 16 and shares a fraction μ of edges connected to the vertices in other clusters, and $1 - \mu$ to the vertices in its own cluster. We call the μ mixing parameter and μ ranges 0 to 0.5 in our experiment. And we adopt the NMI to measure the similarity between the detected partitions and the real ones.

Fig. 3 represents the tests of our algorithm on the LFR

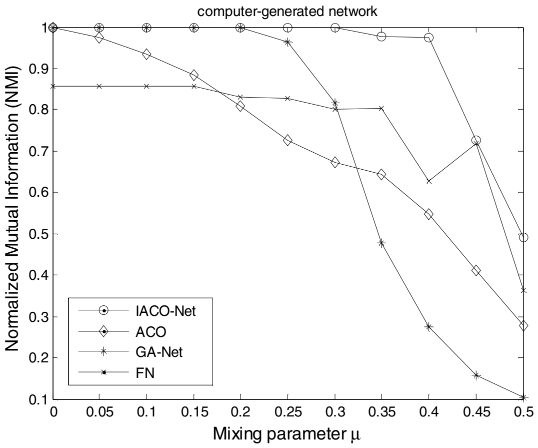


Fig. 3. Experiments on computer-generated network.

benchmark network, and that of ACO, FN and GA-Net as well. Each vertex in the curve responding to the largest NMI of 10 implementations. By observing the curve, all of them have the same trend: when the mixing parameter μ is small, the NMI of each algorithm is high. With an increasement of μ, the structure of the network becomes fuzzy, and the NMI of all the methods decrease drastically. And it is obvious that our algorithm receives a better results in most cases with different value of mixing parameter μ.

C. Results on Real-World Networks

In this part, we will show the experiments of IACO-Net on four undirected real-world networks. They are the Zachary's karate club, the Dolphin social network, the American College football and the Books about US politics. All the structures of the four networks are known. And we compared the results with ACO, FN, and GA-Net on these networks respectively. The results in TABLE I represent the value Q and NMI of each network. The average Q and NMI of all the networks come from the running of each algorithm for 10 times. The maximum Q of each metric is also listed respectively, and the corresponding NMI as well. Specially, in the GA-Net, the coefficient λ corresponds of different number of communities, and $\lambda \in [1,1.5]$ as told in the algorithm. We set $\lambda = 1$ corresponds of a better result after tests on different values of λ.

Zachary's karate club: This network consists of 34 vertices and 78 edges which was compiled by Zachary [33]. By a consecutive observation of a karate club with 34 members in two years, Zachary found the club is separated into two groups ultimately. In our algorithm, we separate each of the group into two smaller groups, and get a modularity of 0.42 which is higher than the other algorithms.

Dolphin social network: This is a network of 62 vertices and 160 edges constructed by David Lusseau [34] and is based on an observation of 62 bottlenose dolphins' behavior during their life in seven years in Doubtful Sound, New Zealand. The edge between two vertices means the two dolphins have a more frequent association statistically. Naturally, the whole colony is divided into two groups, the female group and the male group. The average value Q we obtained on this network is 0.526, also the higher one. However, the value NMI of our algorithm is smaller than that of FN. Actually, FN is a bottom-top and global optimization algorithm which is deficient in local search. That resulting in a local optimum modularity value of 0.495 and a higher NMI value of 0.606.

TABLE I
EXPERIMENTAL RESULTS ON FOUR REAL-WORLD NETWORKS

Network	Metric		IACO-Net	ACO	GA-Net	FN
karate	Avg.	Q	***0.420***	0.419	0.406	0.381
		NMI	***0.687***	0.638	0.637	0.837
	Max.	Q	***0.420***	0.420	0.406	0.381
		NMI	***0.687***	0.687	0.637	0.837
dolphin	Avg.	Q	***0.526***	0.493	0.467	0.495
		NMI	0.567	0.507	0.412	***0.606***
	Max.	Q	***0.529***	0.504	0.490	0.495
		NMI	0.586	0.592	0.450	***0.606***
football	Avg.	Q	***0.602***	0.531	0.598	0.550
		NMI	0.879	0.861	***0.923***	0.654
	Max.	Q	***0.605***	0.555	0.601	0.550
		NMI	0.909	0.907	***0.929***	0.654
polbook	Avg.	Q	***0.517***	0.476	0.490	0.502
		NMI	***0.523***	0.498	0.441	0.534
	Max.	Q	***0.525***	0.501	0.501	0.502
		NMI	***0.565***	0.558	0.477	0.534

American College football: This network including 115 vertices and 613 edges, which represents American football games between Division IA colleges during regular season Fall 2000, and was established by Grivan and Newman [32]. Therefore, a vertex here is a team and an edge represents the game between the two teams. And the network can be parted into 12 groups. We got an average modularity of 0.602 and NMI of 0.879. Due to the limitation of population diversity in GA-Net, the result is also a local optimum.

Books about US politics: This a network of books on politics consists of 105 vertices and 441 edges, discovered by V. Krebs [14]. In this network, a vertex represent a book on American politics and an edge denotes for the connected two books is frequent co-purchased by the same buyer from Amazon.com. Eventually, the books are classified into 3 classes according to the descriptions and reviews of the books. And in our experiments, the maximal modularity is 0.525 and the corresponding NMI is 0.565.

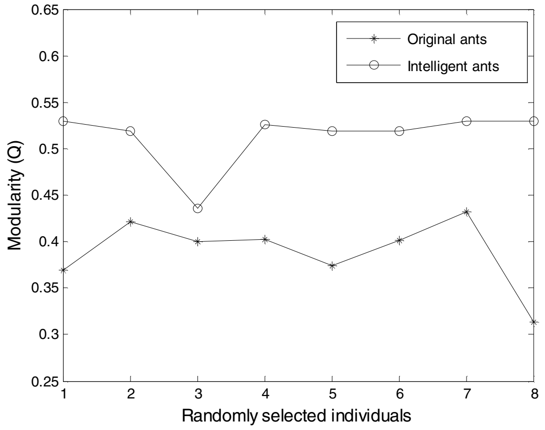


Fig. 4. Comparison between two ant colony generation means, original way and intelligent way (example on Dolphin social network).

In order to show the influence of intelligenct ants during the generation, we randomly choose two group of solutions generated by the original ant colony and intellgent ant colony in the middle procedure of the run. Fig. 4 shows the modularity of two groups and each group including 8 solutions. Obviously, the intelligent ants have a better performance during the iteration.

V. CONCLUSIONS

In this paper, we focus on the modification of the ant colony optimization for community detection problem, and introduce an intelligent ant with learning ability. This learning based strategy can reveal the underlying community structure of a complex network without knowing the exact number of clusters in advance. It is a fact that the performance of ACO is sensitive to the parameters, an improper setting of each parameter will lead to two extreme opposite situations, premature convergence and too much randomness. Fortunately, the procedure of learning from the global optimal solutions can not only avoid redundant computations during the convergence process of ACO, but also can make sure the solutions of the next iteration constructed in a stable and definitely convergent way. The self-learning phase can help the good individuals to promote their quality in a simple but efficiency way. Finally, the experiments on the computer-generated network and real-world networks confirm the advantage and validity of our approach. And in the future, we will focus on the research of adaptive parameters and the combination with other algorithms to receive a better performance of ACO.

REFERENCES

[1] Fortunato, Santo. "Community detection in graphs," *Physics Reports,* vol. 486, no. 3, pp. 75-174, 2010.

[2] Newman, M. E. J. "Communities, modules and large-scale structure in networks," *Nature Physics*, vol. 8, no. 1, pp. 25-31, 2011.

[3] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. "efining and identifying communities in networks, " *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101 no.9, pp. 2658-2663, 2004.

[4] Newman, Mark EJ, and Michelle Girvan. "Finding and evaluating community structure in networks," *Physical review E,* vol.69, no. 2, p. 026113, 2004.

[5] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 9, p. 09008, 2005.

[6] Lancichinetti, Andrea, and Santo Fortunato. "Limits of modularity maximization in community detection," *Physical Review E*, vol. 84. No.6 p. 066122, 2011.

[7] 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, p. 036109, 2008.

[8] M.G. Gong, L.J. Ma, L., Q.F. Zhang, and L.C. 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.

[9] L. Angelini, S. Boccaletti, D. Marinazzo, M. Pellicoro, and S. Stramaglia, "Identification of network modules by optimization of ratio association," arXiv preprint cond-mat/0610182, 2006.

[10] Wei, Y-C., and C-K. Cheng. "Ratio cut partitioning for hierarchical designs," Computer-Aided Design of Integrated Circuits and Systems, *IEEE Transactions on* vol. 10, no. 7, pp. 911-921, 1991.

[11] Newman, M. E., "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no.6, p. 066133, 2004.

[12] R. Guimera, M. Sales-Pardo, and L.A.N. Amaral, "Modularity from fluctuations in random graphs and complex networks," *Physical Review E*, vol. 70, no. 2, p.025101, 2004.

[13] J. Duch, A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, p. 027104, 2005.

[14] M. E. Newman, "Modularity and community structure in networks," Proceedings of the National Academy of Sciences, vol. 103, no. 23, pp. 8577-8582, 2006.

[15] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[16] M. Dorigo, and M. Birattari, "Ant colony optimization," In *Encyclopedia of Machine Learning,* Springer US., pp. 36-39, 2010.

[17] M. Dorigo and L. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 53–66, 1997.

[18] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, "Ant system for job-shop scheduling," *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 34, no. 1, pp. 39-53, 1994.

[19] D. Costa, and A. Hertz. "Ants Can Colour Graphs," *Journal of the Operational Research Society*, vol. 48, pp. 295-305, 1997.

[20] B. Bullnheimer, R.F. Hartl, and C. Strauss, "An improved ant System algorithm for thevehicle Routing Problem," *Annals of Operations Research*, vol. 89, pp. 319-328, 1999.

[21] J.Z. Ji, X.J. Song, C.N. Liu, and X.Z. Zhang, "Ant colony clustering with fitness perception and pheromone diffusion for community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*. vol. 392, no. 15, pp. 3260-3272, 2013.

[22] L. B. Romdhane, Y. Chaabani, and H. Zardi, "A Robust Ant Colony Optimization-based Algorithm for Community Mining in Large Scale Oriented Social Graphs," *Expert Systems with Applications*. vol. 40, no. 14, pp. 5709-5718, 2013.

[23] H.H. Chang, Z.R. Feng, and Z.G. Ren, "Community detection using Ant Colony Optimization," *In Evolutionary Computation (CEC), IEEE Congress on*, pp. 3072-3078, 2013.

[24] T. Stützle and H. H. Hoos, "Max-min ant system," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.

[25] J. Handl, and J. Knowles, "An evolutionary approach to multiobjective clustering," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 1, pp. 56-76, 2007.

[26] A. P. Engelbrecht, "Fundamentals of computational swarm intelligence," Chichester: Wiley, vol. 1, 2005.

[27] T. Stützle and H. H. Hoos, "Max-min ant system and local search for the traveling salesman problem," in *IEEE International Conference on Evolutionary Computation*, pp. 309–314, 1997.

[28] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," In *Parallel Problem Solving from Nature–PPSN X, Springer Berlin Heidelberg*, vol. 5199, pp. 1081-1090, 2008.

[29] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p.066133, 2004.

[30] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment,* vol. 2005, no. 9, p. 09008, 2005.

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

[32] M. Girvan, M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821-7826, 2002.

[33] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452-473, 1977.

[34] David Lusseau, Karsten Schneider, Oliver J. Boisseau, Patti Haase, Elisabeth Slooten, and Steve M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of longlasting associations," *Behavioral Ecology and Sociobiology,* vol. 54, no. 4, pp. 396–405, 2003.