Physica A 392 (2013) 3260–3272

Contents lists available at SciVerse ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Ant colony clustering with fitness perception and pheromone diffusion for community detection in complex networks

Junzhong Ji [a,*], Xiangjing Song [a], Chunnian Liu [a], Xiuzhen Zhang [b]

[a] College of Computer Science and Technology, Beijing University of Technology, Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing, 100124, China

[b] School of Computer Science and Information Technology, RMIT University, Australia

HIGHLIGHTS

- We propose an ant colony clustering approach to discover network communities.
- Each ant uses a new fitness function to percept local environment in the grid.
- Ants employ a pheromone diffusion model to realize information exchange.
- The method combines the local optimization with global intelligence.

ARTICLE INFO

Article history:
Received 10 October 2012
Received in revised form 31 March 2013
Available online 11 April 2013

Keywords:
Complex network
Community structure detection
Ant colony clustering
Fitness perception
Pheromone diffusion model

ABSTRACT

Community structure detection in complex networks has been intensively investigated in recent years. In this paper, we propose an adaptive approach based on ant colony clustering to discover communities in a complex network. The focus of the method is the clustering process of an ant colony in a virtual grid, where each ant represents a node in the complex network. During the ant colony search, the method uses a new fitness function to percept local environment and employs a pheromone diffusion model as a global information feedback mechanism to realize information exchange among ants. A significant advantage of our method is that the locations in the grid environment and the connections of the complex network structure are simultaneously taken into account in ants moving. Experimental results on computer-generated and real-world networks show the capability of our method to successfully detect community structures.

1. Introduction

Nature, society, and many technologies can be modeled by numerous networks that involve many important and complicated interactions among individuals [1], such as Biological networks, Social networks, Collaboration networks, and Web networks. These networks are often called complex networks. These complex networks are found to divide naturally into communities, which are groups of nodes such that the nodes within a group are much more connected to each other than to the rest of the network. In recent years, community structure discovery has been one of the most popular research areas along with its applicability to a wide range of disciplines [2–5]. For example, cell biologists use the community structure in protein interaction networks to make sense of signal transduction cascades and metabolism, to research the inherent relationships between cellular functions and biochemical events in this area. Computer scientists are mapping the Internet and the WWW into different communities to discover topic information from Web pages and potential relationships in

* Corresponding author. Tel.: +86 010 67396568.
E-mail address: jjz01@bjut.edu.cn (J. Ji).

0378-4371/$ – see front matter
http://dx.doi.org/10.1016/j.physa.2013.04.001

hyperlinks. Epidemiologists follow transmission networks through which viruses spread and try to find how to stop the spread of the virus by analyzing the transmission network structure, and so on. That is, communities are interesting and fundamental because they often correspond to functional units and reflect the in-homogeneity or locality of the topological relationships between the nodes of target systems. Thus, community structure mining is important for analyzing topological structures, comprehending network functions, recognizing hidden patterns, and forecasting the future behaviors of complex systems.

With the development and popularity of complex networks, various network clustering algorithms to mine community structures have been proposed [4,6–20]. In Ref. [21], Fortunato gave a comprehensive and in-depth summary for the methods and algorithms of community detection in graphs from different aspects. Although special strategies adopted are different, most of the algorithms are mainly divided into two basic categories from the view of the search mechanism: the global optimization approach and local search approach [20]. The first approach poses clustering as a global optimization problem, which employs an objective function to evaluate the network modularity quality and tries to find an optimal clustering result in the whole solution space [6–8,14–17]. In contrast, there are no global optimization objectives in the second approach, and they carry out clustering based on some local heuristic rules [4,9–13], such as edge betweenness, clique percolation, random walk, etc. Moreover, there are some algorithms that use a combination of these two basic methods [18–20], which can get better clustering results and higher time efficiency than each basic approach. For all that, however, how to further improve the detection accuracy, especially how to discover reasonable community structure without prior knowledge, is still a challenging problem. In this paper, we propose an Ant Colony Clustering algorithm based on Fitness perception and Pheromone diffusion for community detection in complex networks (called as ACC-FP), which is also a combination method. ACC-FP first uses a new fitness function to percept local environment and move to more comfortable locations in the grid. Then, in terms of the ants clustering quality at each iteration, it employs a pheromone diffusion model, which can faithfully simulate the volatilization character of pheromone, to realize information exchange among ants and guide ant colony to perform global searches. Experimental results and relevant comparative analyses show that the combination of local perception and global information feedback is effective to achieve high-quality clustering results.

The rest of this paper is organized as follows. Section 2 introduces related research on community detection. Section 3 presents the motivation and the details of the proposed algorithm. Section 4 presents and analyzes the experimental results. Section 5 concludes this paper.

2. Related work

In the past decade, the problem of community detection has attracted much attention in various scientific fields including physics, sociology, and computer science. Many methods employing different strategies have been proposed and applied successfully to some specific complex networks. For instance, Girvan and Newman proposed GN algorithm, which is a method for detecting a community structure by using information about edge betweenness [4]. Subsequently, Newman proposed a fast algorithm for detecting a community structure based on the Q metric [6]. Based on a q-state Potts model, Reichardt and Bornholdt presented a fast community detection algorithm, which can detect overlapping communities without prior knowledge about the number of communities [7]. By means of the eigenvectors of a modularity matrix for the network, Newman proposed a spectral algorithm for community detection, which can acquire higher quality than some competing methods in shorter running times [8]. Radicchi et al. gave two quantitative definitions of community and presented a local algorithm to detect communities, which can efficiently deal with large-scale networks [9]. Gergely et al. used clique percolation to identify densely connected subgraphs in complex networks [10]. Frey et al. employed affinity propagation to pass messages and iteratively partitioned data points into the exemplar clusters [11]. Yang et al. developed a new algorithm called FEC for identifying communities from signed social networks whose idea rested on an agent-based random walk model [12]. In 2012, Faqeeh et al. put forward a new algorithm (called FA in this paper), which employed the eigenvectors of the clumpiness matrix to construct a "projection space" and used some kind of angular distance in this space to define a border line, and then applied this borderline and hierarchical clustering methods to identify the community structure of a complex network [13]. Gong et al. proposed the improved memetic algorithm called iMeme-Net for solving community detection problems, which combined Label Propagation (PGLP) tactic, an Elitism Strategy (ES), and an Improved Simulated Annealing Combined Local Search (ISACLS) strategy to carry out population generation [14].

It is worth noting that there has been an increasing development trend which employed swarm intelligence to detect the community structure of complex networks in recent years. Pizzuti proposed a new algorithm to discover communities in networks by employing Genetic Algorithms (named GA-Net) [15]. The approach introduced the concept of community score to measure the quality of a partition in communities of a network and tried to optimize this quantity by running the genetic algorithm. Liu et al. proposed a communicating community discovery algorithm based on an ant colony clustering model, which employed movement, picking-up and dropping-down operators to perform node clustering in email networks [16]. To reduce clustering computational costs without loss of solution quality, Sadi et al. used ant colony optimization (ACO) techniques to find cliques in a network and assigned these cliques as meta-nodes, and then employed a traditional algorithm to find community memberships on the reduced graph [17]. To further optimize the approach, authors subsequently used the snowball sampling method to generate these subgraphs and ran the ACO-based clique finding technique on each one in parallel [18]. Taking the Q metric as an objective function, Zhang et al. proposed a mining community method in dynamic social networks, which used a clustering center initialization, the pheromone updating, and heuristic function guiding

strategies to achieve the clustering solution [19]. In Ref. [20], Jin et al. proposed a new ant colony optimization algorithm named ACOMRW, whose basic idea is the progressive strengthening of within-community links and the weakening of between-community links. At each iteration, a Markov random walk model is taken by ants as the heuristic rule; all of the ants' local solutions are aggregated to a global one through clustering ensemble, which is then used to update a pheromone matrix. Gradually this converges to a solution where the underlying community structure of the complex network will become clearly visible.

In essence, community detection in a complex network is a node clustering problem. As a kind of nature inspired algorithm, Ant clustering has shown its ability to produce a low cost, fast, and reasonably accurate solution to the complex clustering problem. For instance, Chen et al. first proposed an ant sleeping model (ASM) and presented the corresponding adaptive ant clustering algorithm in Ref. [22], which mimicked the behaviors of gregarious ant colonies. Later, they extended the work and presented an ant movement (AM) model and an adaptive ant clustering (AAC) algorithm based on the model [23]. Experimental results show that the AAC algorithm based on the AM model is suitable for solving large-scale and complicated clustering problems. However, as far as we know, there is no research of using a similar clustering model to detect the communities from complex networks till date. Moreover, though many computational methods have been available for researchers to detect communities, these methods are not sufficient to completely solve the problem along with the widespread use of complex networks. Hence, both facts inspire us to extend the ASM algorithm and present a new ant clustering algorithm based on ant gregarious and communication behaviors for mining communities in complex networks.

3. The ACC-FP algorithm

3.1. Basic principle

A complex network is typically represented as an undirected graph $G(V, E)$, where V is the set of nodes (or vertices) and E is the set of edges (or links). The community detection in a complex network is essentially a clustering problem, which focuses on detecting densely connected subgraphs $G_1, G_2, \ldots, G_k$ in G, where $G_1, G_2, \ldots, G_k$ satisfy $\cup_{1\leq i\leq k} G_i = G$ and $\cap_{1\leq i\leq k} G_i = \emptyset$. Intuitively, if a partition has the property that within-community edges are dense and between-community edges are sparse, it will be called a well defined clustering result. In this section, we develop a global search algorithm based on an adaptive ant clustering for detection of communities, which uses a new fitness function to percept local environment and employs a pheromone diffusion model to realize information exchange.

In fact, the study of the behavior of real ants has greatly inspired and motivated the developments of ant colony optimization (ACO) and ant colony clustering (ACC). Biologists have discovered that ant nests are near to each other in nature, so ants can take care of each other and collectively combat alien invasions. However, ant nests are not evenly distributed. The ants with similar habits live quite close to each other, whereas the ants with different habits live relatively far away. That is to say, they tend to stay with the species having the same or similar habits. In light of such a living behavior, ants naturally obtain a comfortable environment with the similar gathering, repelling others. The AM model mentioned in Section 2 is the most typical model, which simulates the behavior that ants search for a comfortable position to take a break from the living environment. In light of the similar idea, we model the basic ants and their environment as follows. All ants live in a two-dimensional grid (environment). Each ant is a simple agent who represents a node of a complex network. At the beginning, ants are randomly placed at different locations in a virtual grid. During the evolution process, an ant makes use of the perception of a fitness function to move to a new location or stay in the original location in each iteration. That is, if there is a more comfortable position for the ant, the ant is activated and moves to the new location; otherwise, it will keep the sleeping state and stay still. Ant colony movings at each iteration form a clustering result (solution), whose quality is evaluated and employed to update the pheromone of nodes. To reflect characteristics of the pheromone diffusion, a pheromone diffusion model is applied to the pheromone updating. The evolution process is repeated till all ants find and keep the most comfortable positions, and then the community structure of the complex network is visible in the grid.

3.2. Ant colony perceptions and movements

At each iteration, each solution associated with a clustering result is found by means of ant colony perceptions and movements. The ants respectively move according to the perceptions of the surrounding environments, form into groups eventually, and hence the corresponding nodes are clustered.

3.2.1. Fitness function

Each ant is a simple agent who is able to feel the quality of its local environment, and it frequently tries to search for a more comfortable position for sleeping in its surrounding environment. There are two behavior ways for ants. When an ant is in an inappropriate location, it will actively move around to search for a more suitable position and will not stop until it finds one. Conversely, when an ant is satisfied with its current position, it will keep sleeping. It is easy to see that an ant's perceptual abilities for the environment are very important for such behaviors. To measure an ant's perceptual ability, a

fitness function of an ant i at the iteration t is defined as

$$f_i(t) = \frac{1}{|A|} \sum_{j \in A} \frac{1}{1 + d_{ij}} \tag{1}$$

where A represents the ant i's neighborhood in its living environment (grid), $|A|$ is the number of ants in A, and d_{ij} is the distance between two nodes i and j in the complex network. Since the objects to be clustered are the nodes of a graph, the similarity is often defined in terms of structural equivalence [24]. Thus, d_{ij} in our algorithm is computed as follows:

$$d_{ij} = \sqrt{\sum_{k=1, k \neq i,j}^{N} (a_{ik} - a_{jk})^2} \tag{2}$$

where N is the number of nodes in $G(V, E)$, a_{ij} is an entry of the adjacency matrix, which represents the connectivity relationship between two nodes i and j, and $a_{ij}(i, j = 1, \ldots, N)$ is equal to 1 when the link l_{ij} exists, and zero otherwise. It is obvious that if two nodes i and j are precisely structurally equivalent, the entries in their respective rows of the adjacency matrix will be identical and thus $d_{ij} = 0$. The distance has the properties of a distance metric: (1) $d_{ij} \geq 0$, $\forall i, j$; (2) $d_{ii} = 0$, $\forall i$; (3) $d_{ij} = d_{ji}$, $\forall i, j$.

3.2.2. Moving strategy

If an ant senses more comfortable locations in each iteration, it gets activated and tries to move to a new location in the grid. The new location is randomly selected from the neighborhood of its neighbor nodes in $G(V, E)$ with respect to a probability $P_{ij}(t)$. This probability $P_{ij}(t)$ represents the possibility that the ant i moves to an empty location of the ant j's neighborhood where there is an edge between two represented nodes in $G(V, E)$. It is defined proportionally to an aggregation pheromone factor and a heuristic factor, i.e.,

$$p_{ij}(t) = \begin{cases} \dfrac{[\tau_j(t)]^{\alpha} \cdot [\eta_j]^{\beta}}{\sum_{l \in Neighbor(i)} [\tau_l(t)]^{\alpha} \cdot [\eta_l]^{\beta}}, & j \in Neighbor(i) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$\tau_j(t) = \sum_{k \in A} \frac{\tau_{k \to j}(t)}{d_{kj}} \tag{4}$$

$$\eta_j = \begin{cases} |c(i, j)|, & \sum_{l \in Neighbor(i)} |c(i, l)| \neq 0 \\ d(j), & \text{otherwise} \end{cases} \tag{5}$$

where $\tau_j(t)$ is the quantity of aggregation pheromone laying on the node j at the time t (t is the number of iterations), A represents the node j's neighborhood in the grid, $\tau_{k \to j}(t)$ is the aggregation pheromone on the node j imposed by the node k's pheromone, and d_{kj} denotes the distance between two nodes k and j in the complex network; η_j represents a local heuristic information, $|c(i, j)|$ is the number of common neighbor nodes for the two linked nodes i and j, $d(j)$ is the degree of the node j; $Neighbor(i)$ is the neighbor node set of the node i in the complex network, and the parameters α and β determine the relative importance of pheromone trail versus heuristic factor for the node j. Thus, the higher the value of $\tau_j(t)$ and η_j, the more possible it is to select the node j as the next position.

3.3. Aggregation pheromone diffusion and updating

Aggregation pheromone is an important carrier for an ant colony to implement swarming intelligence. Generally, the more the number of nodes clustered in the local environment around a node is, the larger the pheromone will be; thus, it can attract more and more ants to move closer to its neighborhood. On the other hand, the aggregation pheromone, as a chemistry substance, volatilizes gradually over time, i.e., the pheromone can diffuse around. To simulate this phenomenon, we present a pheromone diffusion model and its corresponding updating formula. The basic idea is to take into account the pheromone influences among close locations in the grid when an ant colony searches a feasible solution. Namely, there are coupling effects among adjacent pheromone diffusion fields; the closer the distance between two locations, the stronger the coupling action is, and yet the farther the distance, the weaker the coupling action is. Thus the pheromone updating needs to perform decoupling compensates for the pheromone intensity of adjacent locations.

3.3.1. Pheromone diffusion model

Let $\tau_k(t)$ be the pheromone trail on a node k laid by the ant colony. We can give the pheromone diffusion model based on the real distance between two nodes in the grid. Fig. 1 shows the sketch map of the pheromone diffusion; Fig. 1(a) presents the relationship between the intensity $\tau_i(t)$ and the distance d_r, and Fig. 1(b) gives the simulation method of the pheromone

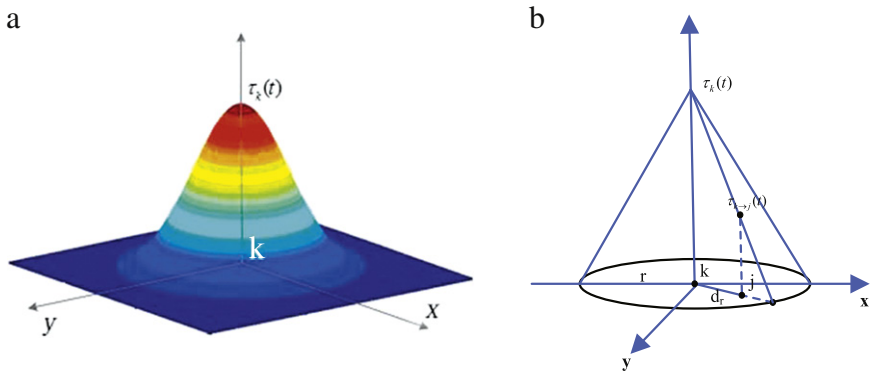

Fig. 1. Sketch map of aggregation pheromone diffusion of an info fountain k and its intensity field. (a) Relationship between the intensity $\tau_i(t)$ and the distance d_r. (b) Simulation method of the pheromone diffusion.

diffusion. To compute the pheromone intensity of nodes in the diffusion field, a cone is employed to simulate the pheromone diffusion model, where the center dot denotes the node k as an info fountain, r is the radius of diffusion area in the grid, the node j locates in the intensity field of the info fountain, and d_r is their distance in the grid. The figure shows that the pheromone diffuses around by taking an info fountain as a center of the diffusion intensity field and that the closer to the info fountain the location, the stronger the field force of the intensity field is. More precisely, the pheromone influence of an info fountain on other nodes will gradually reduce as the distance between nodes becomes long. Therefore, we can give a simple diffusion model:

$$\tau_{k\to j}(t) = \begin{cases} \left(\dfrac{r - d_r}{r}\right) \cdot \tau_k(t), & \text{if } d_r < r \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Based on such a pheromone diffusion model, the aggregation pheromone of all nodes at each iteration can be obtained.

3.3.2. Pheromone updating

Once each solution is formed at each iteration, aggregation pheromone trails of the nodes will be updated. More specifically, the algorithm updates the pheromone intensity for all nodes in light of the profit value of the solution; the formula of pheromone updating is as follows:

$$\tau_j(t+1) = (1-\rho)\tau_j(t) + \triangle\tau_j(t, t+1) \tag{7}$$

where $\tau_j(t+1)$ represents the pheromone trail of the node j used in the iteration $(t+1)$, $0 < \rho \leq 1$ is a coefficient which represents pheromone evaporation, and the $\triangle\tau_j(t, t+1)$ represents the pheromone increment that ants deposited on the node j when the iteration t ends, which depends on the change of the local living environment and the profit value of the solution. In our algorithm, the quality of a solution is evaluated by the average fitness value of all ants, and it is defined by

$$S(t) = \frac{1}{N}\sum_{i=1}^{N} f_i(t) \tag{8}$$

where N is the number of the ant colony. Thus, $\triangle S(t)$ can be denoted as the evolutionary effect of the tth iteration, described as follows:

$$\triangle S(t) = S(t) - S(t-1). \tag{9}$$

Apparently, if $\triangle S(t) > 0$ in the evolutionary process, it means that the clustering result becomes better, vice versa. Meanwhile, the fitness value of each node also shifts along with the change of the global result, which is denoted as follows:

$$\triangle f_j(t) = f_j(t) - f_j(t-1). \tag{10}$$

Based on such global and local changes, $\triangle\tau_j(t, t+1)$ can be computed by the following equations:

$$\triangle\tau_j(t, t+1) = \begin{cases} \displaystyle\sum_{k\in A}(1-\rho)\tau_{k\to j}(t) + Q, & \triangle f_i(t) > 0 \text{ and } \triangle S(t) > 0 \\ \max\left(\displaystyle\sum_{k\in A}(1-\rho)\tau_{k\to j}(t) - Q, 0\right), & \triangle f_i(t) < 0 \text{ and } \triangle S(t) < 0 \\ \displaystyle\sum_{k\in A}(1-\rho)\tau_{k\to j}(t), & \text{otherwise} \end{cases} \tag{11}$$

where $Q > 0$ is a control parameter for the pheromone change. From the descriptions in Eq. (11), it is not hard to see that the incremental quantity of pheromone trail represents a trade-off between the global evaluation of a solution and the local fitness of a node. There are three cases: (1) when the clustering result becomes better and the ant moves to a better position, the incremental quantity of the corresponding node is enhanced; (2) when the clustering result becomes worse and the ant moves to a worse position, the incremental quantity of the corresponding node is softened; (3) in other instances, the incremental quantity only makes use of the pheromone impacts of the local environment. Apparently, this strategy strengthens the effect of good solution while weakening the influence of inferior solution, and fully reflects the idea of swarming intelligence.

3.4. Adaptive adjusting of the fitness threshold value

To switch an ant state, our algorithm gives an initial fitness threshold value $\mu(0)$ at the beginning. At each iteration t, when $f_j(t) < \mu(t)$, the ant is activated and moves according to the moving strategy; otherwise, the ant keeps the sleeping state. If $\mu(t)$ is too small, then most of the ants sleep so that the solution evaluation may remain stagnant. Conversely, most ants move continuously, which also affects the convergence of the algorithm. Thus, $\mu(t)$ is an important parameter which partly determines the algorithm's solving abilities.

To simplify the parameter's selection, a method to self-adaptively adjust the threshold of ant's movement is presented. The main idea is as follows: if the clustering result gradually becomes better by m iterations, we reduce the value of $\mu(t+m)$ to make less ants move and perform fine-tuning for the result along with the good trend. In contrast, if the clustering result is getting worse, we increase the value of $\mu(t+m)$ to make more ants move and increase the scope of the search. The adaptive adjusting rule is defined as

$$\mu(t+m) = \begin{cases} 0.9\mu(t), & \text{if } S(t+m) \geq S(t) \\ 1.1\mu(t), & \text{otherwise.} \end{cases} \tag{12}$$

3.5. Algorithm description and complexity analysis

Summing up the above ideas, ACC-FP can simply be described as follows. In the initialization phase, ants (nodes) are randomly placed at different locations in a virtual grid. A fitness function is employed to percept the comfort levels of the locations. According to the state of comfort levels or the pheromone accumulation, ants decide whether or not to move to more comfortable locations in each iteration, which will result in a clustering solution. Based on the quality of the solution, a pheromone diffusion model is applied to update the pheromone of nodes. The evolution process is iterated till ACC-FP converges to an approximate optimal solution. The algorithm works as shown in Fig. 2.

Let the maximum number of a node degree be k_1 in the complex network, and the maximum neighborhood number of a location be k_2 in the grid. In the initialization phase, computing distances and the number of common neighbors for all pairs of nodes is time-consuming, whose time complexity is $O(k_1 \cdot N)$. In the ant clustering phase, the time complexities of the 2.1 step, the 2.2 step and the 2.3 step are $O((k_1 + k_2) \cdot N)$, $O((k_2 + 1) \cdot N)$ and $O(1)$ (can be negligible), respectively. In the end, the time complexity of the output step is $O(|C|)$. Thus, the overall complexity of ACC-FP is about $O(k_1 \cdot N) + O(T((k_1 + 2k_2) \cdot N + |C|))$. Because most complex networks are small-world and scale-free networks, $k_1 \ll N$. On the other hand, we use the direct neighborhood of a location for the smaller network, so $k_2 = 8$. The double neighborhood (including direct and indirect relations) of a location is only employed for the larger network where $k_2 = 24$; thus $k_2 \ll N$. Moreover, the number of communities $|C|$ is also far less than N. Therefore, the time complexity of ACC-FP can be decreased to $O(T \cdot k' \cdot N)(k' \ll N)$, which is better than typical algorithms with $O(T \cdot N^2)$. As most complex networks are sparse graphs, this will be very efficient.

4. Empirical study

In order to quantitatively analyze the performance of ACC-FP, we use both computer-generated and real-world networks to perform our empirical study. First, we analyze main parameter influences by an empirical testing method in this algorithm. Then, our algorithm is compared with GN [4], Fast Newman (FN) [6], FEC [12], FA [13], iMeme-Net [14], and ACOMRW [20] on three real-world networks, which are all known and competitive network clustering algorithms. Finally, we test the performance of ACC-FP on many computer-generated networks to further illustrate the validity of the approach. Our algorithm is coded using Java.

4.1. Test networks and evaluation metrics

To test ACC-FP, we have performed our experiments over three real-world networks and many computer-generated networks. First, we applied it to three widely used real-world networks with a known community structure. They are the well known Karate Club network [25], Bottlenose Dolphin network [26], and American College football network [4]. The Karate Club network was constructed by Zachary [25], and it is a network of friendship which has 34 members of a karate

Algorithm: ACC-FP.

Input: Graph G(V, E): a complex network;
Output: C: the set of communities;
1.Initialization:
Set various parameters T, N, $\tau(0)$, and threshold value m, $\mu(0)$, ϖ, δ;
* T: maximum number of iterations, m: step length of fitness threshold value *
* N: Number of ant colony (nodes in G(V,E)), $\mu(0)$: initial fitness threshold value *
* $\tau(0)$: initial pheromone value, ϖ: cycle threshold value, δ: pheromone threshold value *
Put randomly each ant at exclusive site on a two-dimensional grid;
Let initial states of ants be the sleeping state and pheromone trail of ants be $\tau(0)$;
Compute distances and number of common neighbors for all pairs of nodes in G(V, E);
For t=0 to T
2. Ants clustering:
{ **2.1 Ant colony perception and movements :**
For i=1 to N
{ Compute ant's fitness $f_i(t)$ according to Eq.(1) and Eq.(2);
If $f_i(t) < \mu(t)$ or (t$\geq \varpi$&$\tau_i(t) \leq \delta$) then * uncomfortable or pheromone too little passing some cycles *
{ Let ant i be active state;
Compute each transfer probability $P_{ij}(t)$ in light of Eq.(3) to Eq.(5);
Select a target ant (node) k from Neighbor(i) by means of roulette;
Move the ant i to a empty place in the ant k's neighborhood, and let the ant i be sleeping state; } }
2.2 Aggregation pheromone diffusion and updating:
For i=1 to N
Calculate the aggregation pheromone for all nodes in light of Eq.(6) to Eq.(11);
2.3 Adaptive adjusting of fitness threshold value:
If (t mod m == 0) then
Adjust the fitness threshold value according to Eq.(12);
3. Output:
If (the same solution obtained in m iterations) then
Return separate communities each other for the complex network.}

Fig. 2. The ACC-FP algorithm.

club as nodes and 78 edges representing friendship between members. Due to a leadership issue, the club splits into two distinct groups. A simple un-weighted version of the network is used in our experiment. The Bottlenose Dolphin network is a community of Bottlenose dolphins living in Doubtful Sound and New Zealand, and it was compiled by Lusseau from the observation of dolphins' behavior during seven years [26]. There are 62 dolphins and edges are set between network members that are seen together more often than expected by chance. The network is split naturally into two large groups, and the number of ties is 159. The American College Football network [4] comes from the United States college football. The network represents the schedule of Division I games during the 2000 season. Nodes in the graph represent teams and edges represent the regular season games between the two teams they connect. The teams are divided into 12 conferences (communities). The teams on average played 4 inter-conference matches and 7 intra-conference matches; thus teams tend to play between members of the same conference. The network consists of 115 nodes and 616 edges grouped in 12 communities.

Second, we adopt some random networks with a known community structure, which have been used as benchmark datasets for testing complex network clustering algorithms [27]. This kind of random network is defined as *LFR*(N, k, γ, ϕ, φ), where N is the number of nodes in a network, k is the average degree of nodes, γ is the exponent of the degree distribution, ϕ is the exponent of the community size distribution, and φ is a mixing parameter which is used to control the ratio between the degree of intra-communities of a node and its total degree.

To compare the clustering accuracy of different algorithms more fairly, we adopt two widely used accuracy measures, which are Fraction of Vertices Classified Correctly (FVCC) [6] and Normalized Mutual Information (NMI) [28]. The FVCC is a simple measure to evaluate the clustering accuracy, while the NMI is adopted to estimate the similarity between the true partitions and the detected ones. The Normalized Mutual Information is a similarity measure proved to be reliable by Danon et al. [28]. Given two partitions A and B of a network in communities, let C be the confusion matrix whose element C_{ij} is the number of nodes of community i of the partition A that are also in the community j of the partition B. The NMI $I(A, B)$ is defined as

$$I(A, B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log(C_{ij}N/C_{i\cdot}C_{\cdot j})}{\sum_{i=1}^{C_A}(C_{i\cdot}\log(C_{i\cdot}/N)) + \sum_{j=1}^{C_B}(C_{\cdot j}\log(C_{\cdot j}/N))} \tag{13}$$

where C_A (C_B) is the number of groups in the partition A (B), $C_{i\cdot}$ ($C_{\cdot j}$) is the sum of the elements of C in row i (column j), and N is the number of nodes. If $A = B$, $I(A, B) = 1$. If A and B are completely different, $I(A, B) = 0$. A larger value of NMI represent a greater similarity between A and B.

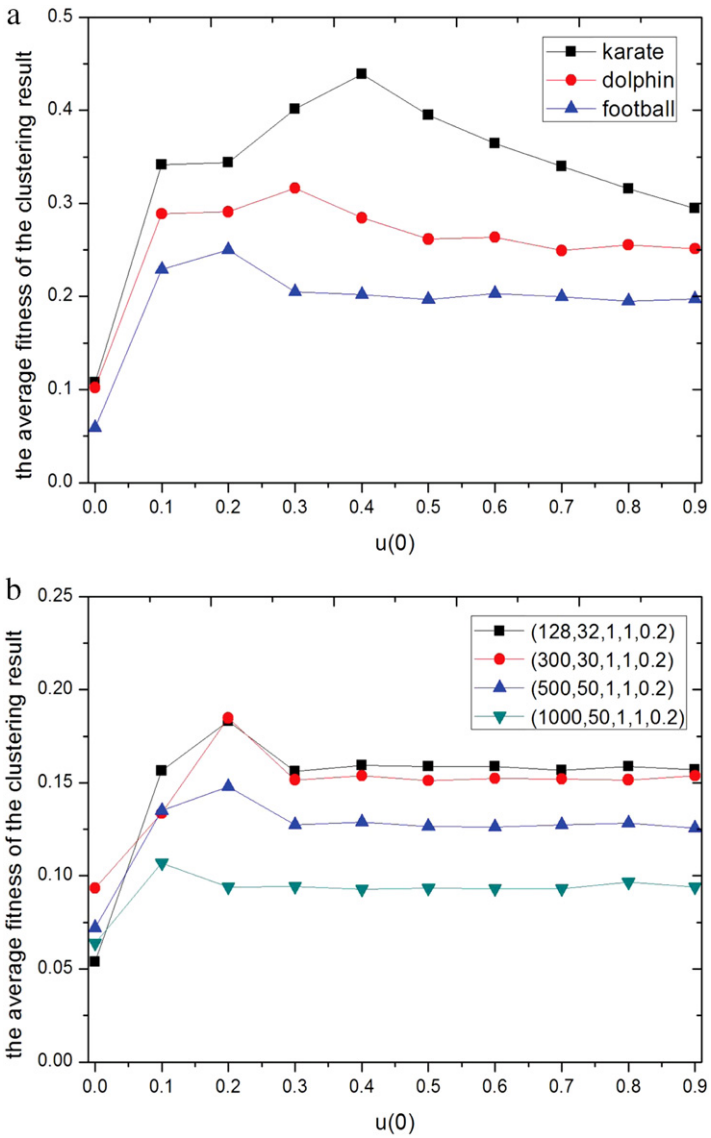

Fig. 3. The effects of $\mu(0)$ parameters on performance. (a) Three different real-world networks. (b) Four different scale computer-generated networks.

4.2. Parameter analyzing and setting

There are more parameters such as T, N, m, $\tau(0)$, ϖ, δ, $\mu(0)$, Q, ρ, α, β in this algorithm, which denote iteration number limitation, ant colony size, step length of updating fitness threshold value, initial pheromone value, cycle threshold value, pheromone threshold value, initial fitness threshold value, pheromone increment, the pheromone evaporation coefficient, and the weights for the pheromone trail and for the heuristic information, respectively. However, it is fortunate that most of the parameters are easy to be determined. For example, $T = 100$, $N = |V|$, $m = 5$, $\varpi = 10$, $\delta = 10$, $\tau(0) = 5$ (which mainly depends on the number of average common neighbors), Q=2, $\rho = 0.6$, $\alpha = 1$, and $\beta = 2$, in general. However, the setting of the parameter $\mu(0)$ is very difficult and important. The value of $\mu(0)$ is directly related to the movement of the ant colony in the evolutionary process and thus has a significant impact on the solving accuracy and efficiency. In this subsection, we focus on studying the effect of $\mu(0)$ by a series of experiments. Fig. 3 shows the results, in which the y-axis denotes the average fitness of the clustering results and the x-axis denotes $\mu(0)$. As we can see from Fig. 3(a), our algorithm obtains the best results when $\mu(0)$ is respectively set to 0.4, 0.3, and 0.2 for karate, dolphin and football networks. As for four different computer-generated networks such as (128, 32, 1, 1, 0.2), (300, 30, 1, 1, 0.2), (500, 50, 1, 1, 0.2), and (1000, 50, 1, 1, 0.2), $\mu(0)$ is respectively set to be 0.2, 0.2, 0.2, and 0.1 to achieve the best performances as shown in Fig. 3(b). Because different networks have different topological characteristics, our algorithm obtains the best results at different values of $\mu(0)$.

Table 1
Compare ACC-FP with some algorithms on three real-world networks.

Algorithms	Karate network			Dolphin network			Football network		
	NMI (%)	FVCC (%)	Num. of Com.	NMI (%)	FVCC (%)	Num. of Com.	NMI (%)	FVCC (%)	Num. of Com.
GN	57.98	97.06	5	44.17	98.39	13	87.89	83.48	10
FN	69.25	97.06	3	50.89	96.77	5	75.71	63.48	7
FEC	69.49	97.06	3	52.93	96.77	4	80.27	77.39	9
ACOMRW	100	100	2	88.88	98.39	2	92.69	93.04	12
FA	100	100	2	88.88	98.39	2	92.42	93.04	12
iMeme-Net	100	100	2	100	100	2	86.2	94.78	12
ACC-FP[a]	100	100	2	100	100	2	93.84	93.04	12
ACC-FP[b]	96.69	100	2.3	88.46	99.03	2.6	90.14	90.17	11.5
ACC-FP[c]	81.62	100	3	72.86	98.39	4	87.03	86.96	12

[a] The best results obtained over 10 runs.
[b] The average results obtained over 10 runs.
[c] The worst results obtained over 10 runs.

4.3. Comparative evaluations on real-world networks

To demonstrate the strength of the ACC-FP method, we compare it with the six competing methods: GN, FN, FEC, ACOMRW, FA, and iMeme-Net on three real-world networks. For each network, we compute the FVCC and NMI on the basis of the best results reported by the authors. Because the ACC-FP is a random optimization method, we run it 10 times with the best parameter setting and compute its average, the best, and the worst performances over these 10 runs.

Table 1 shows the detailed comparative results of the various algorithms on the three different datasets, respectively. For each detection method, we have listed the NMI measure, FVCC measure, and the number of communities detected (number of communities). The results clearly show the very good performance of ACC-FP with respect to other approaches. In fact, on the Karate network, ACC-FP can obtain the best result 7 out of 10 times which are 100% respectively on NMI and FVCC measures, clearly higher than 57.98% and 97.06% of GN, 69.25% and 97.06% of FN, 69.49% and 97.06% of FEC, and competes with those of ACOMRW, FA, and iMeme-Net. Even in the worst case, ACC-FP also gets 81.62% and 100%, which are superior to those of GN, FN, and FEC. On the Dolphin network, ACC-FP obtains 3 times the best results over 10 runs. Its best NMI and FVCC values are both 100%, which are the same as results of iMeme-Net and better than those of other algorithms. The worst result of our algorithm on FVCC is higher than both of FN and FEC, equal to the result of GN, ACOMRW, and FA, and is only worse than that of iMeme-Net. Though our algorithm is inferior to ACOMRW, FA, and iMeme-Net in terms of the NMI metric, it is still well ahead of GN, FN, and FEC. On the American Football network, our algorithm obtains an average NMI of 90.14% over the 10 runs, with a worst value of 87.03% and a best value of 93.84%. The average NMI value of our algorithm is only inferior to 92.69% of ACOMRW and 92.42% of FA, and superior to that of other algorithms. Moreover, our algorithm obtains an average FVCC of 90.17% over the 10 runs, with the worst value of 86.96% and the best value of 93.04%. At worst, our algorithm is able to get the result better than that of GN, FN, and FEC. In other words, our algorithm outperforms the GN, FN, and FEC algorithms and has comparative performance with the ACOMRW, FA, and iMeme-Net algorithms on three real-world networks in terms of the NMI and FVCC metrics.

Fig. 4 gives the best clustering results of ACC-FP on three real-world networks with a known community structure. The Karate Club network is a simple structure of the network; thus, it is easy for ACC-FP to get the best result.

For the dolphin network, though most of the algorithms recently proposed always put the SN89 into a wrong category when the whole network is divided into two large categories, ACC-FP can classify it into its proper category. The reason is that many algorithms always use the connection relationship such as the number of common neighbors between two nodes as a heuristic rule to induct the clustering process, which may result in a wrong decision when the connection relationship does not provide useful information for the clustering process. Fortunately, the ACC-FP algorithm has a good moving mechanism to overcome the shortcoming. We take the SN89 as an example to explicate our moving strategy. The SN89 is connected to only two nodes: node SN100 which belongs to the first category and node Web that belongs to the second category. Neither of the two nodes has common neighbor with SN89; thus the membership to one of the two communities is indistinguishable for many algorithms without adding information. Our algorithm not only uses a combination of heuristic information which includes the common neighbor information and the degree of the node, but also employs the aggregation pheromone to reflect the clustering situation. Since the degree of the SN100 is 7 while that of the Web is 9, it is difficult for the heuristic information to play a role in ants moving. However, the quantity of aggregation pheromone laying on the two nodes will induce ants to move in the right direction; thus, the ACC-FP algorithm can get the exact partitioning of dolphins.

For the American College Football network, our algorithm divides it into 12 clusters which correspond exactly to 12 communities, where only 3 clusters such as Conference USA, Independents, and Sun Belt have some minor errors and 8 nodes including 28, 36, 42, 59, 63, 90, 97 and 110 are grouped into wrong communities. There are two cases for wrong partitioning: (1) a node has no connection with other nodes in its own community, which means there is no regular game in the located conference. Both nodes 28 and 110 belong to this case, so ants incorrectly divide them into Sun Belt community and Western Athletic community, respectively. (2) the number of connections in the same community is less than that of connections between communities. For example, the Sun belt teams (59, 63 and 97) played nearly as many games against

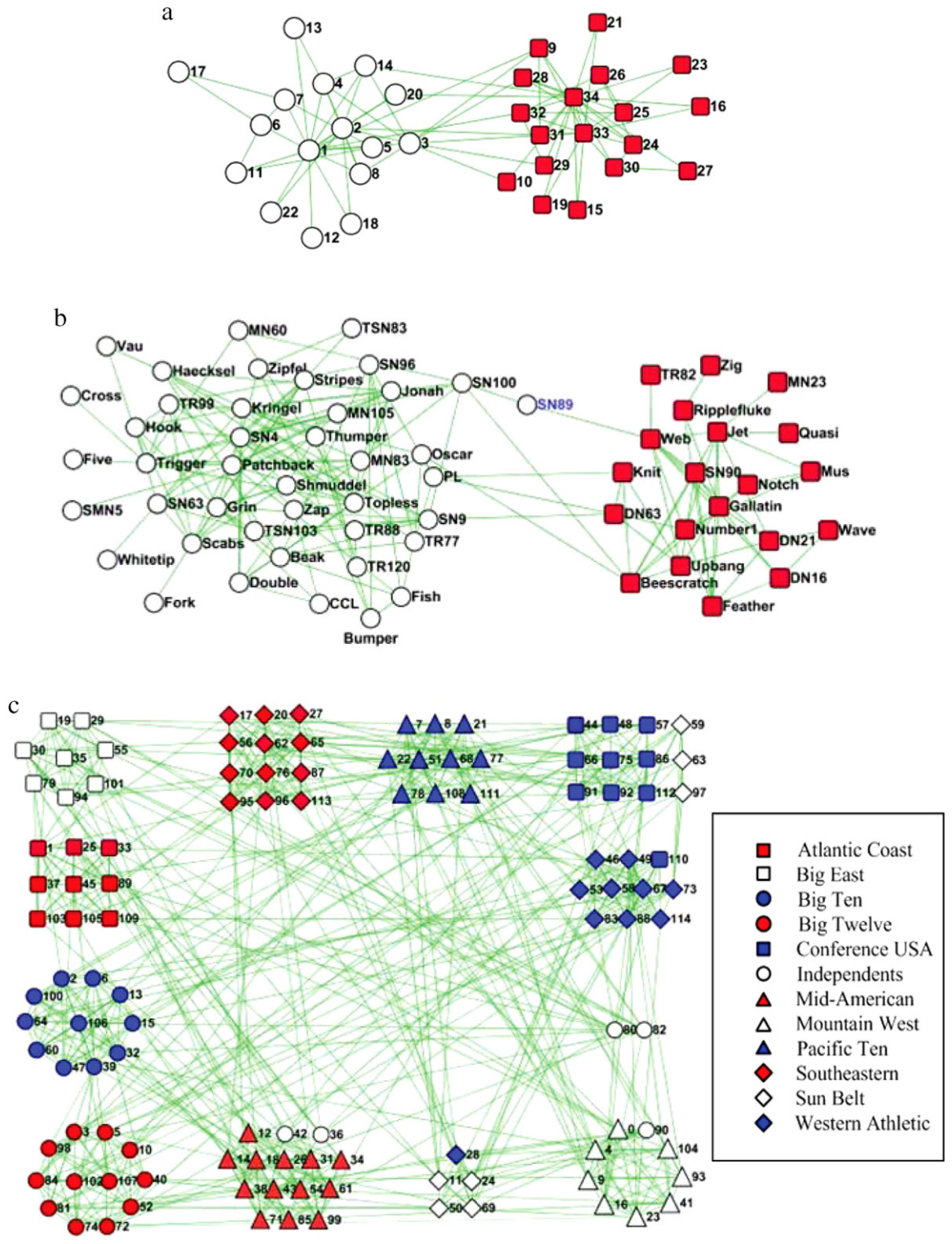

Fig. 4. The best clustering results of ACC-FP on three real-world networks. (a) The Zachary's Karate Club network. (b) The Bottlenose Dolphin network. (c) The American College Football network.

Conference USA teams as they did in their own conference. The Independents teams (36, 42 and 90) also played quite a large fraction of their interconference games against Mid-American teams or Mountain West teams. Naturally, both cases have a common feature, that is, the network structure genuinely does not correspond to the conference structure. In such

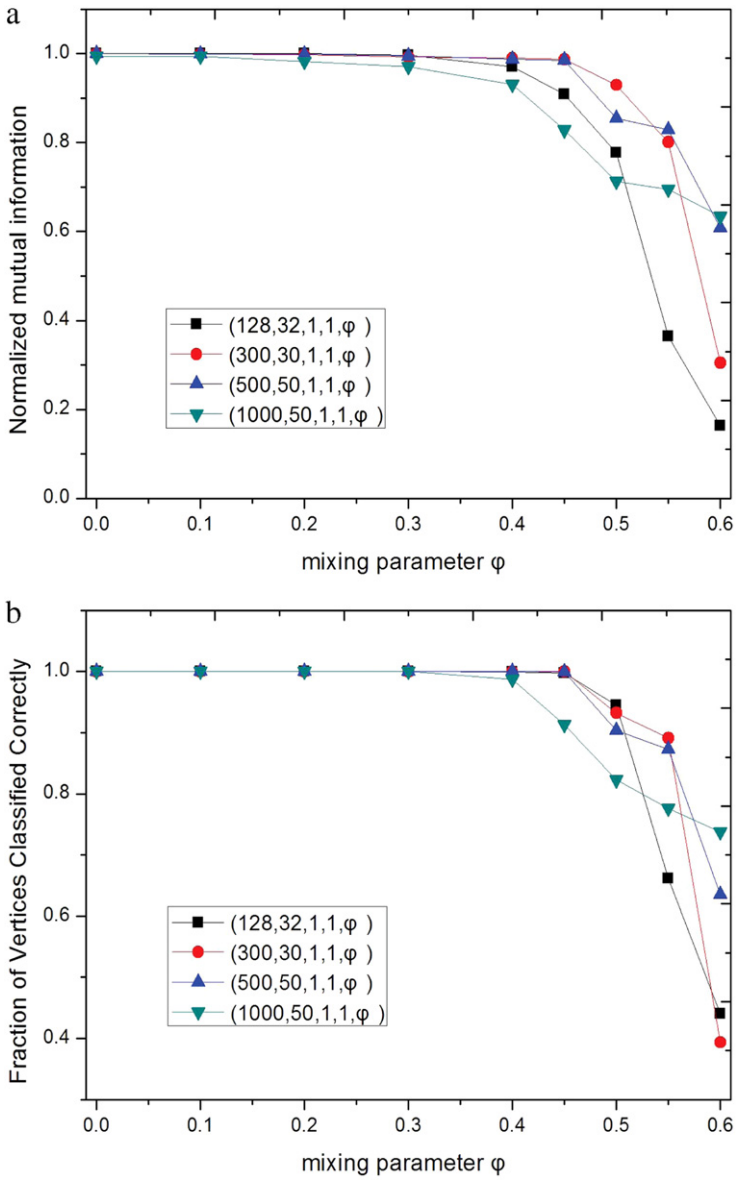

Fig. 5. The NMI and FVCC performances averaged over the 10 runs for different values of φ and N. (a) The NMI performances. (b) The FVCC performances.

two cases, our algorithm also fails to correctly cluster these nodes just like some state-of-the-art algorithms. However, the ACC-FP algorithm performs remarkably well in all other cases.

4.4. Performance on computer-generated networks

As a further test on the algorithm ACC-FP, we applied it to the computer-generated networks with a known community structure. These networks may have different topological properties than the real-world ones. Here we use four types of networks in different scales, $LFR(N, k, \gamma, \phi, \varphi)$, to perform our experiments. According to the meaning of the mixing parameter φ, the degree of intra-communities of a node is larger than that of inter-communities of a node when φ is less than 0.5, and it is obvious that as φ increases from zero, community structures of networks become more diffused and the generated networks present greater and greater challenges to community identification algorithms. For each scale of networks, we generated 9 different networks for the value of φ ranging from 0 to 0.6 and used the NMI and FVCC metrics to measure the similarity and the accuracy between the true communities and the detected ones. For each network, we computed the average performances over 10 independent runs.

Fig. 5 shows the NMI and FVCC performances averaged over the 10 runs for different values of φ and N. Fig. 5(a) points out that until $\varphi \leq 0.4$ the ACC-FP algorithm is successful in detecting the true communities in almost 100% of cases for

three types of networks with $N = 128$, 300 and 500. Even for the networks with $N = 1000$, the ACC-FP algorithm still gets at least 90% of the true communities. When $\varphi \geq 0.45$, the NMI of the ACC-FP algorithm gradually goes worse; however, it should be noted that the fall degree of larger scale networks ($N = 500$ or 1000) is less than that of the smaller networks ($N = 128$ or 300) when $\varphi \geq 0.5$, which shows that the ACC-FP algorithm has better stability when it solves the larger scale networks. As is shown in Fig. 5(b), we also get the similar variation on FVCC performance, and the ACC-FP algorithm can still correctly classify almost 100% of nodes into their correct communities for types of networks when $\varphi = 0.4$. On the whole, though the higher the number of interconnections, the more indistinguishable the network structure is because communities are mixed with each other; the ACC-FP algorithm is still able to effectively identify the hidden communities, especially for large-scale networks.

No matter on the real-world networks or on the computer-generated networks, the results obtained show the capability of our algorithm to effectively deal with community identification in networks, and the algorithm performance is competitive with that of state-of-the-art approaches on the real-world networks.

5. Conclusions

The identification of community structures in complex networks is of great interest because they often reveal unknown relationships between nodes and provide useful information for unknown nodes. However, how to accurately predict community structures by computational methods is still a highly challenging issue. The paper presents an ant colony clustering algorithm with high accuracy for identifying a community in complex networks. The algorithm focuses on the strategy of ant perception and movements and the method of pheromone diffusion and updating, and searches for an optimal partitioning of the network by ant colony movements. The underlying community structure of the complex network will become clearly visible at the end of the algorithm by selectively exploring the search space, without the need to know in advance the exact number of groups. The performance of our algorithm was tested on three real-world networks, as well as on a set of computer-generated networks. Experimental results confirm the validity and advantage of this approach. Future research will aim at combining our method with other methods to improve the quality of results.

Acknowledgments

We would like to thank the anonymous referees for their many valuable suggestions and comments. We thank the authors of the corresponding algorithms for sharing the binary executables of their systems. This work is partly supported by the NSFC research program (60825203), 973 (2011CB302703) and the Beijing Natural Science Foundation (4102010).

References

[1] A.L. Barabási, Scale-free networks: a decade and beyond, Science 325 (5939) (2009) 412–413.
[2] A.L. Barabási, R. Albert, H. Jeong, G. Bianconi, Power-law distribution of the world wide web, Science 287 (5461) (2000) 2115–2115.
[3] R. Albert, H. Jeong, A.L. Barabási, Error and attack tolerance of complex networks, Nature 406 (6794) (2000) 378–382.
[4] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Science 99 (12) (2002) 7821–7826.
[5] R. Guimerà, L.A.N. Amaral, Functional cartography of complex metabolic networks, Nature 433 (7028) (2005) 895–900.
[6] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Physical Review E 69 (6) (2004) 066133.
[7] J. Reichardt, S. Bornholdt, Detecting fuzzy community structures in complex networks with a Potts model, Physical Review Letters 93 (21) (2004) 218701.
[8] M.E.J. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Science 103 (23) (2006) 8577–8582.
[9] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Proceedings of the National Academy of Science 101 (9) (2004) 2658–2663.
[10] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814–818.
[11] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[12] B, Yang, W.K. Cheung, J.M. Liu, Community mining from signed social networks, IEEE Transactions on Knowledge and Data Engineering 19 (10) (2007) 1333–1348.
[13] A. Faqeeh, K.A. Samani, Community detection based on the "clumpiness" matrix in complex networks, Physica A 391 (7) (2012) 2463–2474.
[14] M.G. Gong, Q. Cai, Y.Y. Li, J.J. Ma, An improved memetic algorithm for community detection in complex networks, in: IEEE World Congress on Computational Intelligence, Brisbane, Australia, 2012, pp. 1–8.
[15] C. Pizzuti, GA-Net: a genetic algorithm for community detection in social networks, in: The 10th International Conference on Parallel Problem Solving from Nature, Dortmund, Germany, 2008, pp. 1081–1090.
[16] Y. Liu, J.Y. Luo, H.J. Yang, L. Liu, Finding closely communicating community based on ant colony clustering model, in: The 2010 International Conference on Artificial Intelligence and Computational Intelligence, Sanya, China, 2010, pp. 127–131.
[17] S. Sadi, Ş. Etaner-Uyar, Ş. Gündüz-Öğüdücü, Community detection using ant colony optimization techniques, in: The 15th International Conference on Soft Computing, Brno, Czech Republic, 2009, pp. 206–213.
[18] S. Sadi, Ş. Öğüdücü, A.Ş. Uyar, An efficient community detection method using parallel clique-finding ants, in: The 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain, 2010, pp. 1–7.
[19] N. Zhang, Z. Wang, Community mining in dynamic social networks-clustering based improved ant colony algorithm, in: The 6th International Conference on Computer Science & Education, SuperStar Virgo, Singapore, 2011, pp. 472–476.
[20] D. Jin, D.Y. Liu, B. Yang, C. Baquero, D.X. He, Ant colony optimization with markov random walk for community detection in graphs, in: The 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Shenzhen, China, 2011, pp. 123–134.
[21] S. Fortunato, Community detection in graphs, Physics Reports 486 (3) (2010) 75–174.
[22] L. Chen, X.H. Xu, Y.X. Chen, An adaptive ant colony clustering algorithm, in: The Third International Conference on Machine Learning and Cybernetics, Shanghai, China, 2004, pp. 1387–1392.

[23] X.H. Xu, L. Chen, An adaptive ant clustering algorithm, Journal of Software 17 (9) (2006) 1884–1889.
[24] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.U. Hwang, Complex networks: structure and dynamics, Physics Reports 424 (4–5) (2006) 175–308.
[25] W.W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33 (4) (1977) 452–473.
[26] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behavioral Ecology Sociobiology 54 (4) (2003) 396–405.
[27] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Physical Review E 78 (4) (2008) 046110.
[28] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics-Theory and Experiment 78 (09) (2005) P09008.