An Enhanced Particle Swarm Optimization Based on *Physarum* Model for Community Detection

Zhengpeng Chen[1], Fanzhen Liu[1], Chao Gao[1,2,3](✉), Xianghua Li[1], and Zili Zhang[1,4]

[1] School of Computer and Information Science, Southwest University, Chongqing 400715, China
cgao@swu.edu.cn
[2] Potsdam Institute for Climate Impact Research (PIK), 14473 Potsdam, Germany
[3] Institute of Physics, Humboldt-University zu Berlin, Berlin 12489, Germany
[4] School of Information Technology, Deakin University, Geelong, VIC 3220, Australia

Abstract. Community detection, an effective tool to analyze and understand network data, has been paid more and more attention in recent years. One of the most popular methods of detecting community structure is to find the division with the maximal modularity. However, the modularity maximization is an NP-complete problem. In the field of swarm intelligence algorithm, particle swarm optimization (PSO) has been widely used to solve such NP-complete problem. Nevertheless, premature convergence and lower accuracy limit its performance in community detection. In order to overcome these shortcomings, this paper proposes a novel PSO called P-PSO for community detection through combining the computational ability of *Physarum*, a kind of slime. The proposed algorithm improves the efficiency of PSO by recognizing inter-community edges based on *Physarum*-inspired network model (PNM). Experiments in eight networks show that the proposed algorithm is effective and promising for community detection, compared with other algorithms.

Keywords: Community detection · PSO · *Physarum* network model

1 Introduction

Complex networks have numerous characteristics, among which the community structure is an important one. Community detection, a powerful tool to discover community structures, has a wide application prospect, like predicting protein functions [1] and analyzing the information dissemination [2].

In the past few decades, a large number of algorithms have been proposed for community detection. They can be classified into optimization algorithm and

Y. Tan et al. (Eds.): ICSI 2017, Part II, LNCS 10386, pp. 99–108, 2017.
DOI: 10.1007/978-3-319-61833-3_11

heuristic algorithm. Meanwhile, a modularity measure Q [3] is proposed to evaluate the quality of community divisions, which has been widely used. It has been proved that swarm intelligence optimization algorithms including particle swarm optimization algorithm (PSO) [4] show their superiority in local learning and global search. Recently, Cai et al. have successfully used greedy discrete particle swarm optimization algorithm (GDPOS) [5] to detect the community structures in a network. However, failing to make full use of prior knowledge of network and generate high-quality initial population, this algorithm does not lead to the good enough performance of global search and relatively high accuracy.

According to the latest reports, a large number of biological experiments have demonstrated that a slime named *Physarum* has an intelligence of solving mazes and constructing efficient and robust networks [6,9]. Meanwhile, the *Physarum*-inspired Mathematical Model (PM) has been proposed by Tero et al. [7], which has been used for optimizing the heuristic algorithms [8]. Thus, a *Physarum*-inspired network model (PNM) is proposed for initializing the PSO based on the PM model, which is utilized to distinguish inter-community edges from intra-community edges. Furthermore, we attempt to optimize the phase of PSO's initialization for higher quality in community detection.

The remaining of this paper is organized as follows: Sect. 2 illustrates the related background and introduces the particle swarm optimization algorithm for community detection. Section 3 proposes the *Physarum*-inspired particle swarm optimization algorithm. Section 4 reports the experiments in eight real-world networks and the comparisons with state of the art algorithms. Section 5 concludes this paper.

2 Related Work

2.1 Community Detection

A network can be composed of nodes and edges, in which nodes usually stand for members and edges represent relationships between members. Let $G = (V, E)$ denote a network, where V and E are the aggregations of nodes and edges, respectively. Aiming at dividing the nodes in a network into different communities, community detection results in that nodes across communities are sparsely connected, while nodes within a community are relatively densely connected. Under the premise that a community is a subset of V and n_c is defined as the number of communities, a community division is a set of communities, $C_i \subset G, C = \{C_1, C_2, \ldots, C_{n_c}\}$, where $C_i \neq \varnothing$, $\bigcap_{i=1}^{n_c} C_i = \varnothing$, $\bigcup_{i=1}^{n_c} C_i = G$.

In this paper, the proposed fitness function is the widely used modularity (normally denoted as Q) [3]. The Q function can be written as Eq. (1), where $|V|$ and $|E|$ are the number of nodes and edges of a network, respectively; A is the adjacency matrix of a network and $A_{ij} = 1$ if there exists an edge between node i and j; k_i is the degree of node i, and $\delta(i, j) = 1$ if the nodes i and j are in the same group, otherwise $\delta(i, j) = 0$. Without the loss of generality, we assume that the better division corresponds to the higher Q value. Therefore, the community detection can be transformed into an optimization problem formulated as Eq. (2).

$$fit(\cdot) = Q = \frac{1}{2|E|}\sum_{i,j}^{|V|}(A_{ij} - \frac{k_i \cdot k_j}{2|E|})\delta(i,j) \tag{1}$$

$$C^* = \arg\max_{C} Q(C,G) \tag{2}$$

2.2 PSO for Community Detection

Derived from the social behavior seen in some animal populations, like fish school and birds flock, PSO is a type of swarm intelligence algorithm proposed by Eberhart and Kennedy in 1995 [4]. The concise framework, simple principle and fast convergence make PSO a popular algorithm for solving continuous optimization problems. Each particle has a position and velocity vector. The position vector usually stimulates a candidate solution to the optimized problem, and the velocity vector denotes the tendency of position updating. A particle updates its status iteratively according to its own and the other particles' experiences to search for the optimal solution. Here, we take a typical PSO for network clustering, termed GDPSO, as an example to introduce the basic parts of PSO for community detection.

Particle representation: Considering that the community detection is a discrete optimized problem, we have to redefine the particle positions. One position vector represents a network division and the position vector of the particle i is defined as $X_i = \{x_i^1, x_i^2, \ldots, x_i^n\}$, where $x_i^j \in [1, n]$ is an integer.

In such definition, x_i^j is called a label identifier standing for the community the node j belongs to. If $x_i^j = x_i^k$, then node j and k belong to the same community. Not only is this coding scheme easy to decode, but also it can determine the number of the communities after division directly. As a result, the computational complexity will be reduced. The coding scheme of the particle is shown in Fig. 1.

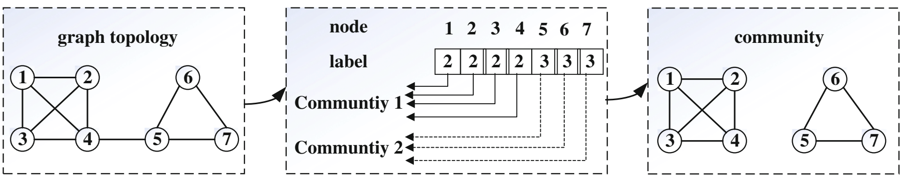


Fig. 1. The coding scheme of the particle in GDPSO. Each particle is coded as a string of integers, which represents the label identifier of the corresponding node.

Particle-status-updating rules: The operation of updating status must be redefined under the discrete background in order to make GDPSO practicable for community detection. The updating rules are put forward as follows:

$$V_i = \omega V_i \oplus (c_1 r_1 (Pbest_i \ominus X_i) + c_2 r_2 (Gbest_i \ominus X_i)) \tag{3}$$

$$X_i = X_i \otimes V_i \tag{4}$$

In the above equations, the $Pbest_i = \{pbest_i^1, pbest_i^2 \ldots, pbest_i^D\}$ and $Gbest_i = \{gbest_i, gbest_i \ldots, gbest_i\}$ are the i^{th} particle's best personal position and the best global position of the swarm, respectively; the inertia weight ω, the learning factors c_1 and c_2 are set typical values of 0.7298, 1.4961 and 1.4961; the r_1 and r_1 are random numbers ranging from 0 to 1.

In Eq. (3), $\ominus$ is defined as an XOR operator. Provided with two velocity vectors $V_1 = \{v_1^1, v_1^2, \ldots, V_1^n\}$ and $V_2 = \{v_2^1, v_2^2, \ldots, v_2^n\}$, $V_1 \oplus V_2 = V_3 = \{v_3^1, v_3^2, \ldots, v_3^n\}$ is a velocity vector with a detailed operation shown as follows:

$$\begin{cases} v_3^i = 0, & rand(0,1) \geqslant \dfrac{1}{1+e^{-(v_1^i + v_2^i)}} \\ v_3^i = 1, & rand(0,1) < \dfrac{1}{1+e^{-(v_1^i + v_2^i)}} \end{cases} \tag{5}$$

In Eq. (4), given an old position $X_{old} = \{x_{old}^1, x_{old}^2, \ldots, x_{old}^n\}$ and a velocity $V = \{v_1, v_2, \ldots, v_n\}$, $X_{old} \otimes V = X_{new} = \{x_{new}^1, x_{new}^2, \ldots, x_{new}^n\}$ is a position vector whose element is defined as follows:

$$\begin{cases} x_{new}^i = x_{old}^i, & if \quad v_i = 0 \\ x_{new}^i = \arg\max_j \varDelta Q(x_{old}^i, j | j \in L_i), & if \quad v_i = 1 \end{cases} \tag{6}$$

where $L_i = \{l_1, l_2, \ldots, l_k\}$ is the set of label identifiers of node $i's$ neighbors. The $\varDelta Q$ is calcluated using the following equation:

$$\varDelta Q(x_{old}^i, j | j \in L_i) = fit(X_{old} | x_{old}^i \leftarrow j) - fit(X_{old}) \tag{7}$$

In general, each node chooses the community identifier which contributes to the largest increase or the smallest decrease of Q value based on its neighbors.

Mutation: GDPSO implements the mutation operation so as to preserve diversity and avoid falling into local optima. The procedure can be depicted as follows: generating a random number between 0 and 1; for each node in a network, if the random number is smaller than the mutation probability pm, assigning its label identifier to all of its neighbors.

3 *Physarum*-inspired PSO for Community Detection

3.1 The *Physarum*-based network mathematical model

In this paper, PM model is modified into *Physarum*-based network model (PNM) which could be used to recognize the intra-community edges in a network. The key mechanism of PM model is the feedback system between the fluxes and conductivities of tubes based on the Posieuille flow.

First, let $Q^t_{i,j}$, $D^t_{i,j}$, $L_{i,j}$ and p^t_i stand for the flux, the conductivity, the length of $e_{i,j}$ and the pressure of v_i at time step t, respectively. The relationship among these parameters can be represented as Eq. (8). Second, according to the Kirchhoff's law formulated in Eq. (9), the pressure and fluxes can be obtained by solving such equations at each iteration step. Third, $Q^t_{i,j}$ feeds back to $D^t_{i,j}$ based on Eq. (10), and as iteration step t is completed, the iteration step $t+1$ repeats the above procedures on the basis of the data iteration step t returns. Finally, as such positive feedback continues, a highly efficient network is generated [7].

$$Q^t_{i,j} = \frac{Q^t_{i,j}}{L_{i,j}}|p^t_i - p^t_j| \tag{8}$$

$$\sum_i Q^{t-1}_{i,j} = \begin{cases} I_0, & \mathit{if}\ v_j\ \mathit{is\ an\ inlet} \\ -I_0, & \mathit{if}\ v_j\ \mathit{is\ an\ outlet} \\ 0, & \mathit{others} \end{cases} \tag{9}$$

$$D^t_{i,j} = \frac{(Q^t_{i,j} + D^{t-1}_{i,j})}{k} \tag{10}$$

PNM is based on the *Physarum*-inspired Mathematical Model (PM), whose major modification is the scheme of choosing inlets/outlets in each iteration. In such model, a vertice is chosen as an inlet, while the others are chosen as outlets. Namely, Eq. (9) is modified as Eq. (11), where D and L are known. Given a certain inlet and outlet, a set of equations based on Eq. (11) can be obtained. By solving such equations, we get p_i of node i, where i ranges from 1 to $|V|$. Besides, every vertice is chosen as the inlet once in each iteration step of PNM. When v_i is chosen as the inlet, a local conductivity matrix denoted as $D^t(i)$ is calculated based on the feedback system. Eventually, after all local conductivity matrices are obtained, the global conductivity matrix is updated by the average of $D^t(i)$ based on Eq. (12).

$$\sum_i \frac{Q^{t-1}(i)_{i,j}}{L_{i,j}}|p^t_i - p^t_j| = \begin{cases} -I_0, & \text{if } v_j \text{ is an inlet} \\ \dfrac{-I_0}{|V|-1}, & \text{others} \end{cases} \tag{11}$$

$$D^t = \frac{1}{|V|}\sum_i^{|V|} D^t(i) \tag{12}$$

3.2 *Physarum*-Inspired Network Model for Community Detection

Taking advantage of PNM, we roughly distinguish the inter-community edges from intra-community through conductivities. Then, we adopt PNM optimize initialization generating a high-quality initial solution and accelerating convergence.

We can obtain a matrix D through PNM, and suppose that node i has a neighbor set $L(i) = \{l_1, l_2, \ldots, l_k\}$ and let $label(i)$ be the community label which node i belongs to. First, for each node i, we initialize $label(i)$ as i. In addition, we assume that $\Omega_i = \{label(j)|j \in L(i)\ and\ D_{i,j} < (1 - R\%) * D_{max}\}$ includes the community labels of neighbors of node i. Namely, the top $R\%$ conductivities $D_{i,j}$ denote that the edges between node i and j are inter-community edges. Then, each node randomly selects an element from Ω_i as its new label.

For the next step, the label propagation is utilized to optimize preliminary initial solution further. Each node determines its community label based on the labels of its neighbors. We assume that each node in the network chooses to join the community with the largest number of its neighbors, which can be represented as Eq. (13), where $\delta(i,j)$ is 1, if node i and j belong to the same community, otherwise $\delta(i,j)$ is 0. This step is executed $iters$ times where $iters$ is the number of propagation iteration. For a clear expression, with a prefix (i.e., P−) added to the original GDPSO algorithm for distinction, the novel algorithm is denoted as P-PSO. The detailed process of P-PSO is shown in Algorithm 1.

$$label(i) = \arg\max_{r} \sum_{j \in L(i)} \delta(label(j), r) \tag{13}$$

Algorithm 1. The framework of P-PSO

Input: An adjacent matrix A and the label propagation iterations: $iters$
Output: The community division of a network
1. Calculating the conductivity matrix D;
2. Initializing the population that each node has unique label in each particle;
3. **for** each $particle \in population$ **do**
4.   **for** $i = 1 : nodes$ **do**
5. $label(i) \leftarrow choose\ a\ label\ randomly\ from\ \Omega_i$;
6.   **for** $j = 1 : iters$ **do**
7.   **for** $j = 1 : nodes$ **do**
8. $label(i) \leftarrow formula$ (13);
9. Evaluating the fitness of population and initializing the $Gbest$ particle;
10. **while** not satisfy the terminal condition **do**
11.   **for** each $particle$ **do**
12.   Updating particle status, see Sect. 2.2 for more information;
13.   Operateing mutation on particle, see Sect. 2.2 for more information;
14.   Evaluating the fitness of particle and updating the $Pbest$ particle;
15.   Evaluating the fitness of swarm and updating the $Gbest$ particle;

4 Experiments and Results

All experiments are executed in the same environment to enable fair comparisons between our algorithm and other algorithms including GDPSO [5], IACO-Net [12] and PNGACD [13]. All results are averaged over 30 repeated runnings in order to eliminate fluctuation. There are two popular metrics for evaluating the performance of community detection: the modularity Q and normalized mutual information (NMI) [10].

4.1 Results on Benchmark Networks

Some experiments are carried out in the GN benchmark network proposed by Lancichinetti et al. [11]. α denotes the mixing parameter which controlls the proportion of links within and out of a community. We test all algorithms in eleven computer-generated networks with the value of α ranging from 0 to 0.5.

As shown in Fig. 2, when the mixing parameter is no larger than 0.1, all algorithms except PNGACD can discover the correct communities ($NMI = 1$). With the mixing parameter increasing, the IACO-Net fails to detect the true partitions. For $\alpha = 0.4$, P-PSO and GDPSO still obtain $NMI = 1$. When α is larger than 0.4, the NMI of GDPSO decreases more quickly than the proposed P-PSO. The experiments in the GN benchmark networks prove that P-PSO is feasible for community detection.

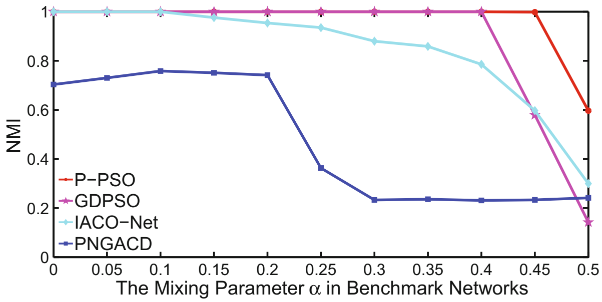


Fig. 2. The experimental results from the GN benchmark networks.

4.2 Results on Real-World Networks

Table 1 shows the structural characteristics of eight real-world networks used in our experiments for evaluating the performance of our proposed method.

Figure 3 shows the results that some experiments are implemented to verify the robustness of P-PSO in the four networks. It can be concluded that P-PSO has a better stability than that of GDPSO. Table 2 reports the maximal and

Table 1. Networks used in this paper. *Clusters* stands for the number of communities in standard divisions, in which "–" means that the standard division is non-existent.

Network	Nodes	Edges	Clusters	Network	Nodes	Edges	Clusters
Karate	34	78	4	Dolphins	62	159	2
Polbooks	105	441	3	Football	115	613	12
Lesmis	77	254	–	Adjnoun	112	425	–
SFI	118	200	–	Celegans	297	1540	–

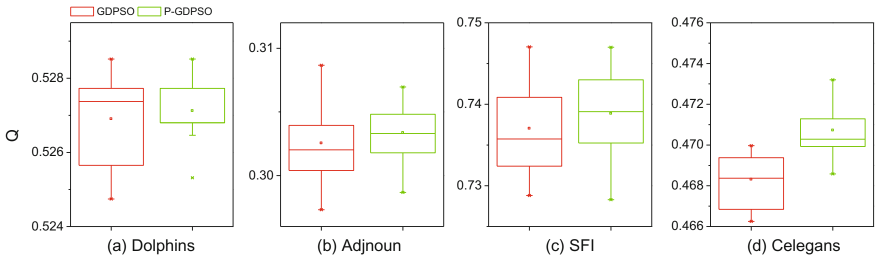


Fig. 3. The average Q of the final iteration in four real-world networks. The upper and lower ends of whiskers represent the maximum and minimum of Q, and the vertical height of the box ranges from the first and the third quartiles. Besides, the small square and band inside the box denote the average and median of Q, respectively. These box charts demonstrate that P-PSO is inclined to a better robustness in community detection.

Table 2. The test results for the Football, SFI and Celegans in terms of Q_{max} and Q_{avg}

Network	Football		SFI		Celegans	
	Q_{max}	Q_{avg}	Q_{max}	Q_{avg}	Q_{max}	Q_{avg}
P-PSO	0.6046	0.6046	0.7470	0.7389	0.4732	0.4717
GDPSO	0.6046	0.6046	0.7470	0.7370	0.4707	0.4685
IACO-Net	0.6032	0.5817	0.1940	0.1969	0.3733	0.3622
PNGACD	0.5973	0.5856	0.7457	0.7400	0.2914	0.2903

mean values of Q in other real-world networks. Results show that P-PSO is substantially better than the compared algorithms.

Figure 4 reports the dynamic average modularity with the increment of iteration. The optimized algorithm P-PSO has a higher growth rate than the original GDPSO at the initial phase. The difference between them becomes smaller with the increment of iteration, and yet P-PSO converges faster than GDPSO. Above all, P-PSO shows a superiority in Q value during the whole iteration process.

Figure 5 shows the community divisions in Polbooks and Football. In Fig. 5(a), the geometric figures denote the real communities and the colors denote communities detected by P-PSO. Due to the context of books, some books are connected more closely and form smaller communities, which disorganizes the original divisions in the real world. In terms of the Football network, the positions are denoted as the real division and the colors mean five communities in the division of P-PSO. Each node represents a football team in the real world, and an edge stands for a game they have together. The marked circle emphasizes the main difference between the detected communities by P-PSO and the real communities.

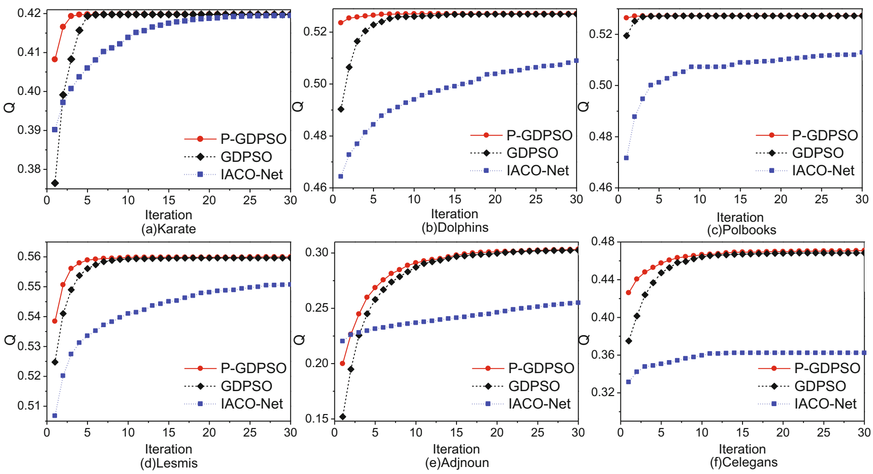
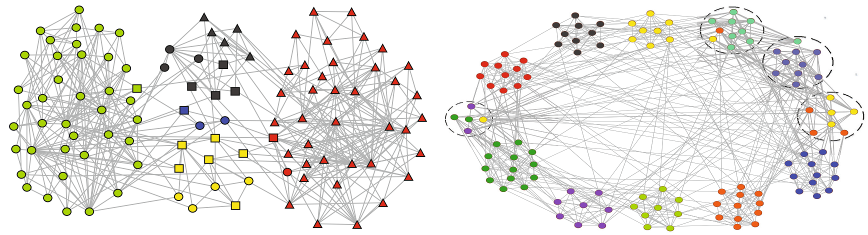


Fig. 4. The dynamic Q with the increment of iteration. The results show that the proposed algorithm can accelerate the convergence, compared with GDPSO and IACO-Net.



(a) P-PSO in Polbooks (b) P-PSO in Football

Fig. 5. The visualizations of community divisions in two networks

5 Conclusion

The research about community detection is helpful for us to analyze the basic characteristics of networks. Taking advantage of the *Physarum* network model (PNM) and greedy discrete particle swarm optimization algorithm (GDPSO), we propose a particle swarm optimization algorithm (P-PSO). The experimental results in eight real-world networks demonstrate that P-PSO shows a better ability in optimizing the initial solution and can obtain effective and promising results than other state of the art algorithms.

Acknowledgments. Zhengpeng Chen and Fanzhen Liu contributed equally to this work and should be considered as co-first authors. This work is supported by the National Natural Science Foundation of China (Nos. 61402379, 61403315), Fundamental Research Funds for the Central Universities (No. XDJK2016A008, XDJK2016B029, XDJK2016E074), CQ CSTC (cstc2015gjhz40002).

References

1. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
2. Weng, L., Menczer, F., Ahn, Y.Y.: Virality prediction and community structure in social networks. Sci. Rep. **3**, 2522 (2013)
3. Newman, M.E.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**, 8577–8582 (2006)
4. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of 1995 IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Press, New York (1995)
5. Cai, Q., Gong, M., Ma, L.: Greedy discrete particle swarm optimization for large-scale social network clustering. Inf. Sci. **316**, 503–516 (2015)
6. Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D.P., Fricker, M.D., Yumiki, K., Kobayashi, R., Nakagaki, T.: Rules for biologically inspired adaptive network design. Science **327**, 439–442 (2010)
7. Tero, A., Kobayashi, R., Nakagaki, T.: A mathematical model for adaptive transport network in path finding by true slime mold. J. Theor. Biol. **224**, 553–564 (2007)
8. Liu, Y., Gao, C., Zhang, Z., Lu, Y., Chen, S., Liang, M., Tao, L.: Solving np-hard problems with physarum-based ant colony system. IEEE/ACM Trans. Comput. Biol. Bioinf. **14**, 108–120 (2017)
9. Nakagaki, T., Yamada, H., Tóth, Á.: Intelligence: maze-solving by an amoeboid organism. Nature **407**, 470–470 (2000)
10. Danon, L., Daz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. J. Stat. Mech. Theory Exp. **2005**, P09008 (2005)
11. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**, 046110 (2008)
12. Mu, C., Zhang, J., Jiao, L.: An intelligent ant colony optimization for community detection in complex networks. In: 2014 IEEE Congress on Evolutionary Computation(CEC), pp. 700–706. IEEE Press, New York (2014)
13. Gao, C., Liang, M., Li, X., Zhang, Z., Wang, Z.: Network community detection based on the Physarum-inspired computational framework. IEEE/ACM Trans. Comput. Biol. Bioinf. (2016). doi:10.1109/TCBB.2016.2638824