



A new nature-inspired optimization for community discovery in complex networks

Xiaoyu Li¹, Chao Gao² , Songxin Wang³, Zhen Wang², Chen Liu⁴, and Xianghua Li^{2,a}

¹ School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

² School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

³ School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China

⁴ School of Ecology and Environment, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

Received 28 March 2021 / Accepted 12 May 2021 / Published online 7 July 2021

© The Author(s), under exclusive licence to EDP Sciences, SIF and Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract. The community structure, owing to its significant status, is of extraordinary significance in comprehending and detecting inherent functions in real networks. However, the community structures are always hard to be identified, and whether the existing algorithms are based on optimization or heuristics, the robustness and accuracy should be improved. The *physarum* (i.e., slime molds with multi heads) has proved its ability to produce foraging networks. Therefore, we adopt *physarum* so that the optimization-based community detection algorithms can work more efficiently. Specifically, a *physarum*-based network model (pnm), which is capable of identifying inter-edges of the community in a network, is used to optimize the prior knowledge of existing evolutionary algorithms (i.e., genetic algorithm, particle swarm optimization algorithm and ant colony algorithm). The optimized algorithms have been compared with some advanced methods in synthetic and real networks. Experimental results have verified the effectiveness of the proposed method.

1 Introduction

The community structure contributes a lot to the dynamic characteristics and potential functions in a network, and has been applied widely in many aspects [1–3]. Since nodes are densely connected, they are gathered in the same cluster; otherwise, they are resided in different clusters. If some communities are related, then the intact network can be affected. Therefore, communities are detected so that the related communities can be found and displayed [4]. Community structures can reinforce our understanding of networks, which can be used in numerous areas (e.g. engineering, social and biological networks) [5, 6].

Currently, one type of popular method is to maximize one objective such as the modularity [7] and the identifiability of community structure [8]. This type of method is also to maximize multi-objective like kernel k -means or ratio cut [9] to detect communities. Then some evolutionary algorithms [10, 11], such as genetic algorithms (GA) [12], particle swarm optimization algorithms (PSO) [13] and ant colony optimization algorithms (ACO) [14, 15], can be applied in this problem. Nevertheless, since the network structure is too com-

plex to be identified easily, an algorithm desperately requires to be more accurate and economical in computation cost. Based on existing extensive research in the field of evolutionary computation [16], it is manifested that the prior knowledge, as the essential element of evolutionary algorithms, is of great importance for computational cost. Therefore, the motivation of this paper is to improve the computational efficiency of EAs-based algorithms through optimizing the prior knowledge.

Inspired by calculating ability and positive feedback theory following foraging process of *Physarum* [17], a large amoeba-like cell is developed, which is composed of a dendritic network of tube-like pseudopodia. A *Physarum*-inspired optimization strategy has been proposed for providing more valuable prior knowledge of existing evolutionary algorithms and improving the computational efficiency of solving NP-hard problems (e.g., TSP [18] and 0/1 KP [19]). However, the problems below still need to be explored.

1. Can the *Physarum* identify community structures and provide prior knowledge for evolutionary algorithms?
2. Is it possible to construct the *Physarum*-based computing framework to improve the precision and cut the computing expenditure of EAs-based algorithms?

^ae-mail: li_xianghua@nwpu.edu.cn (corresponding author)

This paper assesses the effectiveness of the proposed method with more performance metrics and networks.

2 Related work

2.1 Optimization-based community discovery approaches

The optimization-based community detection is first formulated as some optimization problems. Then, some heuristic algorithms are proposed to solve and optimize such problems. Because of the higher computational efficiency and easy-to-implement characteristic in solving optimization problems, EAs-based algorithms have been used to identify the structure of a network through optimizing the Q [16]. More specifically, the typical EAs-based algorithms, i.e., GA [12]-, PSO [13]- and ACO [14]-based optimization methods, motivated by theories from biological areas and ethological areas, respectively, have united with community discovery issues.

Through imitating Darwin natural selection and genetic theory, community discovery approaches based on GA (e.g., GACD [21] and GLAS [22]) search for the optimizing solution given an objective function. These approaches encode the candidate solution (i.e., the community partitions) into integer strings, named as chromosomes, due to the attribute partition of edge terminals. Genes are the integers in these chromosomes. What's more, the fitness value of every chromosome represents the homologous value in the objective function. In each iteration step, the GA-based algorithms search better chromosomes by both crossover and mutation operations. As these steps go on, it is obvious that a higher fitness value helps chromosomes to live when the iteration operators increase. Finally, GA-based algorithms output the chromosome which has the highest fitness value and decode it to form the greatest community division.

As the representative of the swarm intelligence algorithms, the PSO is also evolution algorithms of a certain kind. Each particle in the PSO can adjust its state according to its historic experience and neighbors for the exploration of solution space. With the continuous update of the population position, the search of algorithms gradually falls in the high-quality solution space. When the algorithm reaches the convergence and the maximal iteration generation, we decode the particle with the global optimal position to acquire the optimal solution. Though the GDPSO discretize the PSO algorithm for community detection [13], its accuracy still limits the development in community detection.

ACO is another typical optimization algorithm motivated by the community behavior of social insect ants [23]. There are two basic elements in ACO: pheromone matrix and heuristic factor [14, 15]. The pheromone matrix provides a chance for each ant to find the community structure all alone. After that, every ant renews the pheromone on spot to notice other ants

about community partition condition on the basis of a pheromone matrix. At each iteration, the pheromone matrix instructs ants to find solutions, and the quality of solutions feeds back to other ants by renewing the pheromone matrix. With the iteration going, these ants can find the best community partitions through the pheromone. Although every ant can find a solution which reflects its local opinion, the whole optimal solution is acquired after every solution being aggregated by clustering [24].

Due to the importance of the prior knowledge of EAs, researchers design hybrid algorithms through combining existing evolutionary algorithms with other heuristic algorithms, aiming to provide more valuable prior knowledge for EAs. For example, to deal with TSP and 0/1 KP, we can develop the computational efficiency of original ACO and GA by optimizing the initialization and renewing the pheromone matrix [18]. Inspired by such strategies, we aim to optimize the existing optimization-based community detection algorithms combined with a new computational model, inspired by the intelligent behaviors of a sort of multi-headed slime molds (i.e., *Physarum*) [17]. Although some preliminary experiments are implemented [16], the detailed formulations and comparisons are not addressed.

2.2 The computational intelligence of *Physarum*

Nowadays, owing to the capacity of searching for optimal solutions of food and establishing networks with great robustness, *Physarum*, a kind of slime mould, has been taken seriously. Some studies have reported that *Physarum* has shown an ability to deal with maze and design network among experiments in biology [17]. Since *Physarum* can accomplish these tasks without concentrated command, the theory has caught plenty of attention. To better grasp and understand the main forging characteristics and mechanisms of *Physarum*, a number of researchers have put forward lots of models to characterize and reappear the forging procedure on the basis of many computational approaches motivated by nature (e.g., artificial immune system [25]).

For example, based on a feedback system, a mathematical model is used to imitate the flux propagation in *Physarum* networks [26]. Based on such a feedback system, the crucial pseudopodia survives and gets stronger while others disappear. According to these features and mechanisms, some complex computational problems (e.g., TSP and 0/1 KP [18]) and real problems in our world (e.g., traffic network optimization [27] and multicast routing problem [28]) can be solved in the field of *Physarum*-based computing. Inspired by the analyses and applications above, this paper tries to combine the information of *Physarum* with prior knowledge of EAs (e.g., GA, PSO and ACO) to improve the robustness and accuracy of existing GA-based, PSO-based and ACO-based community discovery approaches.

2.3 Performance metrics for community detection

For estimating the efficiency and effectiveness of different community detection algorithms, we should determine some performance metrics and criterion for our experiments.

First, the modularity Q is selected as the fitness function and objective function for GA-based, PSO-based and ACO-based community discovery approaches, respectively. The Q , defined in Eq. (1), has been widely used as the optimization objective in many algorithms [15]. A larger value of Q tends to indicate a better division. Such a criterion reflects the inherent features of a network, i.e., the heterogeneous relationships among edges between dense intra-community connections and sparse inter-community connections.

$$Q = \frac{1}{2|E|} \sum_{i,j}^{|N|} \left(A_{i,j} - \frac{d_i d_j}{2|E|} \right) \delta(i,j), \tag{1}$$

where $|E|$ and $|N|$ represent the size of edges and nodes. d_i is the degree of a node n_i . $\delta(i,j)$ denotes whether the connection exists between n_i and n_j in the community division. More specifically, when $\delta(i,j)$ equals to 1, n_i and n_j reside in the same community; otherwise, nodes belong to different communities. $A_{i,j}$ represents the element of A , where $A_{i,j}$ equals to 1 when n_i and n_j are connected.

Second, the other more reasonable quality criterion for community detection is the NMI for networks with known community division [29]. NMI , defined in Eq. (2), measures the similarity between two divisions (i.e., the detected and standard communities) on the basis of a determinate characteristic in reality [5]. If we have both a ground truth S and a discovered community structure D , then a confusion matrix Co between these two divisions is acquired.

Finally, if the ground truth of a network is known, some criteria can help us further estimate the accuracy of results. For example, the fraction of nodes identified correctly ($FVIC$) [35] and the number of detected clusters (dc) in a network can be used to justify whether the divisions contains unreasonable fragments. Thereby, both NMI and $FVIC$ can be applied to estimate the efficiency of community discovery approaches when provided the ground truth.

$$NMI(S,D) = \frac{-2 \sum_{i=1}^{p_S} \sum_{j=1}^{p_D} Co_{i,j} \cdot \log\left(\frac{Co_{i,j} \cdot |N|}{Co_{i \cdot} \cdot Co_{\cdot j}}\right)}{\sum_{i=1}^{p_S} Co_{i \cdot} \cdot \log\left(\frac{Co_{i \cdot}}{|N|}\right) + \sum_{j=1}^{p_D} Co_{\cdot j} \cdot \log\left(\frac{Co_{\cdot j}}{|N|}\right)}, \tag{2}$$

where $Co_{i,j}$ denotes how many nodes are in d_i and s_j in the meantime. p_S and p_D denote the size of clusters in division S and D , separately. $Co_{i \cdot}$ is the total of factors in row i and similarly, $Co_{\cdot j}$ represents total factors in column j . When the detected partition D matches

standard division S , then $NMI(S,D) = 1$, otherwise $NMI(S,D) = 0$.

3 Physarum-inspired framework for detection

3.1 The Physarum-based network mathematic model

3.1.1 The process of solving the maze problem

A maze is represented as a graph $G(N,E)$, in which N denotes a node set and E represents an edge set. We set n_1 as the inlet and $n_{|N|}$ as the outlet. $d_{i,j}$ is the distance between nodes n_i and n_j . It is the main task for the maze problem to search for the $\min(d_{1,i} + d_{i,j} + \dots + d_{k,|N|})$, where $n_i, n_j, n_k \in N$.

Tero et al. formulate the characteristic of “key pipelines key focus”. First, pipelines with inside flux in the biological experiment is presented as the edges in a network. $D_{i,j}$ and $PQ_{i,j}$ are represented as Eq. (3), in which $E_{i,j}$ is the distance of corresponding edges and P_i denotes the pressure of n_i [26]. $D_{i,j}$ and $PQ_{i,j}$ represent the conductivity and quantity of pipe combining nodes n_i and n_j , respectively.

$$PQ_{i,j}^t = \frac{D_{i,j}^{t-1}}{L_{i,j}} |P_i^t - P_j^t|. \tag{3}$$

Then the total flux is assumed as I_0 . According to the conservation law, the inflow I_{in} and the outflow I_{out} are both I_0 at any time. This case is explained as Eq. (4). On the basis of the original setting, the pressure of each node can be calculated by combining Eqs. (3) and (4).

$$\sum_i PQ_{i,j}^t = \begin{cases} I_0, & \text{if } n_j \text{ is an inlet} \\ -I_0, & \text{if } n_j \text{ is an outlet} \\ 0, & \text{others} \end{cases}, \tag{4}$$

PQ^t is updated based on Eq. (3). Before the iteration process, D^{t+1} is affected by PQ^t in Eq. (5), where k is a parameter. The conductivities are reflected in the flux at the next time. The feedback system between cytoplasmic flux and conductivity of tube, which is also critical to PNM, entertains that larger fluxes enhance the conductivity while smaller fluxes reduce the conductivities. Finally, a high efficient network emerges owing to this feedback process.

$$D_{i,j}^{t+1} = \frac{PQ_{i,j}^t + D_{i,j}^t}{k}. \tag{5}$$

3.1.2 The description of PNM

The Physarum-based network model (PNM) establishes a node as an inlet once every time. A quintessential

example is formulated as Eq. (6) that is used to replace Eq. (4). The global conductivity is renewed by computing the mean $D^t(i)$ based on Eq. (7), where $D^t(i)$ is the updated conductivity matrix when n_i is an inlet at t . After the iteration number surpasses the determinate threshold, the loop will end.

$$\sum_i \frac{D_{i,j}^{t-1}}{L_{i,j}} |P_i^t - P_j^t| = \begin{cases} I_0, & \text{if } n_j \text{ is an inlet} \\ \frac{-I_0}{|N|-1}, & \text{others} \end{cases}, \quad (6)$$

$$D^t = \frac{1}{|N|} \sum_{i=1}^{|N|} D^t(i). \quad (7)$$

Algorithm 1 The process of PNM method

In The adjacent matrix A ;
Out The conductivity matrix D ;
 For ($t = 1 : T$)
 For ($i = 1 : |N|$)
 Selecting inlet: n_i ;
 Computing P_i^t on the basis of **Eqs. (3) and (6)**;
 Computing $PQ_{i,j}^t$ on the basis of **Eq. (3)**;
 Computing $D^{t+1}(i)$ on the basis of **Eq. (5)**;
 Updating the D^{t+1} on the basis of **Eq. (7)**;
 Outputting D ;

In view of the computational capability of PNM in identifying the edges between different communities from those inside the same community, this paper proposes a *Physarum*-inspired computing method in the following section to make the optimization-based algorithms more efficient.

3.2 *Physarum*-inspired initialization

By means of PNM method for premised work, we can differentiate the inter-community edges and intra-community ones based on the conductivities. After that, with the help of the recognition PNM, the prior knowledge of optimization-based community detection algorithms can be perfected to make the computation more robust and more accurate. More specifically, we refer to many coding and initialization approaches for better GA-based (i.e., NGACD [12]), PSO-based (i.e., GDPSO [13]) and ACO-based (i.e., ACOC [15], IACONet [14]) community discovery approaches by enhancing premier solutions and heuristic elements, separately. To be distinguished from original algorithms, the P-prefix takes the lead in the new name of optimized algorithms.

3.2.1 Improved GA-based algorithms based on PNM

In this section, we exemplify the NGACD [12], the characteristic GA-based community discovery method, to present the optimized initialization of our proposed framework. First, D is obtained on the basis

of Algorithm 1. Also, DA is generated to stand for features of edges, where $DA_{i,j}$ denotes the property of $e_{i,j}$. In detail, $DA_{i,j} = -1$ only when $e_{i,j}$ is an inter-community edge. Then entire edges are initialized as inter-community edges. Subsequently, central nodes generate from the selection of nodes at random. The DA indicates that the corresponding value of the edge connecting them immediately changes to 1 (i.e., $DA_{i,j} = 1$), unless the edges with the maximal conductivity. Finally, DA is utilized to obtain the initial community structures. The process above needs to be repeated until the number of initial solutions we have established before are acquired.

In P-NGACD (a prefix “P-” is added before the original name), the population is coded as $genome_{|N| \times |N|}$. A $genome_{i,j}$ denotes the character division of edge between nodes n_i and n_j . As for DA , $genome_{i,j} = 1$ when the two nodes come from the same community. Or else, $genome_{i,j}$ equals to -1 . So far, by $genome_{i,j}$, the $genome_{|N| \times |N|}$ is exhibited to depict the character division. Equipped with this coding plan, the convenience for the crossover and mutation method to exchange the character of edges and change a gene bit separately has been improved greatly.

3.2.2 Improved PSO-based algorithms based on PNM

This section takes GDPSO algorithm as an example to exhibit the optimized initialization of our framework. The initialization of GDPSO based on PNM (i.e., P-GDPSO) is concluded as follows.

D is primarily obtained on the basis of Algorithm 1. The next step is to initialize the position vector of each particle, denoted as $X = \{x^1, x^2, \dots, x^{|N|}\}$, where $x^i \in [1, |N|]$ and assign a unique community number for each node. Firstly, each node n_i in the position vector randomly selects one neighbor n_j that ensures the conductivity between n_i and n_j is smaller than the fixed threshold. Then, the label of n_j is assigned to n_i . After that, the iteration is processed until the result converges or the iteration reaches a certain number. If there is more than one label with highest frequency, the final label will be selected among these labels at random. The operation of updating is formulized in Eq. (8).

$$x^i = \arg \max_r \sum_{j \in L_i} \varphi(x^j, r), \quad (8)$$

where L_i is a set of neighbor nodes of n_i . If $x^j = r$, then $\varphi(x^j, r) = 1$; otherwise, $\varphi(x^j, r) = 0$.

$$V_i = \omega V_i \odot (c_1 r_1 (Pbest_i \oplus X_i) + c_2 r_2 (Gbest \oplus X_i)). \quad (9)$$

After the initialization, the fitness of population is evaluated and the global optimal position $Gbest$ is chosen. Then the velocity and position of particles are updated by Eqs. (9) and (10). The operator \oplus performs the XOR operation on two position vectors. $Pbest_i$ is the indi-

vidual optimal solution of the i^{th} particle. The inertia weight ω , learning factor c_1 and c_2 are set to 0.7298, 1.4961 and 1.4961, respectively. r_1 and r_2 are random numbers between 0 and 1.

$$X_i = X_i \otimes V_i, \tag{10}$$

where \odot calculates two velocity vectors. For example, there are two vectors $V_1 = \{v_1^1, \dots, v_1^{|N|}\}$ and $V_2 = \{v_2^1, \dots, v_2^{|N|}\}$. $V_3 = V_1 \odot V_2 = \{v_3^1, \dots, v_3^{|N|}\}$ is defined in Eq. (11).

$$\begin{cases} v_3^i = 0, \text{rand}(0, 1) \geq \frac{1}{1+e^{-(v_1^i+v_2^i)}} \\ v_3^i = 1, \text{rand}(0, 1) < \frac{1}{1+e^{-(v_1^i+v_2^i)}} \end{cases}. \tag{11}$$

The operational rule of \otimes is defined in Eq. (12). If $v_1^i = 0$ and x_1^i keeps unchanged, then the label of n_i remains unchanged. Otherwise, it will be taken by that of its neighbor nodes and the increased modularity $\Delta Q(x_1^j, j|j \in L_i)$ will be calculated. Finally, n_i updates the label as the one which contributes the most to the increased modularity.

$$\begin{cases} x_2^i = x_1^i, & v_1^i = 0 \\ x_2^i = \arg \max_j \Delta Q(x_1^j, j|j \in L_i), & v_1^i = 1 \end{cases}. \tag{12}$$

3.2.3 Improved ACO approaches based on PNM

This section exemplifies the ACOC [15] to demonstrate the optimizing process of ACO by means of PNM. First, a probability matrix $P(i, c_j)$ is used to direct each ant to finish the whole route and each ant is a solution in Eq. (13), which is a community division. $P(i, c_j)$ declares the possibility of n_i residing in C_j . And c_j labels the community C_j .

$$P(i, c_j) = \frac{\eta(i, c_j)^\beta \text{Tau}(i, c_j)^\alpha}{\sum_k \eta(i, c_k)^\beta \text{Tau}(i, c_k)^\alpha}, \tag{13}$$

$$\eta(i, c_j) = \frac{n_{i,c_j}}{\sum_{k=1}^p n_{i,c_k}}, \tag{14}$$

where Tau is renewed by ants at every iteration on the basis of the qualities of solutions. $\eta(i, c_j)$ stands for the size of edges between n_i and nodes in a community C_j . The parameters α and β command the importance of pheromone trail factor versus heuristic factor. We can deduce that n_i belongs to a community C_j if there are more edges connecting n_i and C_j . Therefore, $\eta(i, c_j)$ can be selected as a heuristic factor and measured by Eq. (14) to develop the ability of searching among ants. More specifically, n_{i,c_j} is the normalized result to measure how many edges connecting n_i and nodes in C_j .

Before allocating a community label to a node, we should first compute $P(i, c_j)$ for each n_i . And then, n_i will be labeled by a community label c_j with the top

P on the basis of the probability p_0 . And n_i is randomly labeled on the basis of the roulette with the possibility $1 - p_0$. After that, each ant aims to update the pheromone matrix Tau on the basis of Eq. (15), where ρ denotes how fast the pheromone evaporates and Q represents how good the community partition is. Finally, Eq. (16) is implemented to enhance the effects of better community partitions whose Q values are high.

$$\text{Tau}(i, c_j) = (1-\rho) \cdot \text{Tau}(i, c_j) + 2\rho \cdot Q \cdot \delta(i, c_j), \tag{15}$$

$$\text{Tau}(i, c_j) = (1-\rho) \cdot \text{Tau}(i, c_j) + Q_{top} \cdot \delta(i, c_j). \tag{16}$$

To diversify the solutions and prevent the ant colony methods from the precociousness, this paper implements a random walk method. During such phases, each node is first randomly repartitioned by the community label with the possibility p_m . Also, the node will accept such reassignment only when the modularity value of corresponding solution can be improved. Having finished the local seeking and renewing for Tau , ants cluster to a certain community division owning the top Q . To make the existing ACO-based detection approaches more efficient, we optimize the heuristic factor on the basis of Eq. (17).

$$\eta^*(i, c_j) = \frac{n_{i,c_j}^*}{\sum_{k=1}^p n_{i,c_k}^*}, \tag{17}$$

$$n_{i,c_j}^* = \sum_{k \in C_j} \frac{1}{w_{ik}}, \tag{18}$$

where n_{i,c_j}^* is defined in Eq. (18). w_{ik} indicates the conductivity of $e_{i,k}$, a factor of D generated from Algorithm 1, and c_j represents the label of community C_j . The more edges between n_i and nodes in C_j are, the larger the $\eta^*(i, c_j)$ is, in comparison with $\eta(i, c_j)$.

3.3 The mapping of the PNM parameters

Figure 1 illustrates the mapping of PNM parameters to GA and PSO-based algorithms. First, a network, as shown in Fig. 1a, is mapped by the PNM framework in which each node is denoted as a food source, and each edge is denoted as a pipe as plotted in Fig. 1b. The conductivity D is mapped to the factor predicating properties of edges in a network. More specifically, the conductivity D based on Eqs. (3), (5), (6), (7) when we know the structure of a network, is shown in Fig. 1b. After that, there only exists inter-community edges. Then, some central nodes, such as n_4 and n_5 , are randomly selected in Fig. 1c. If the edges exist between those selected nodes (i.e., $e_{1,4}, e_{2,4}, e_{3,4}, e_{4,5}, e_{5,6}, e_{5,7}$) and their conductivities are lower than a certain threshold, they will be marked as intra-community edges. As shown in Fig. 1c, $e_{1,4}, e_{2,4}, e_{3,4}, e_{5,6}, e_{5,7}$, are marked as intra-community edges. The existing research shows that the top 20% conductivities of edges are that of inter-community.

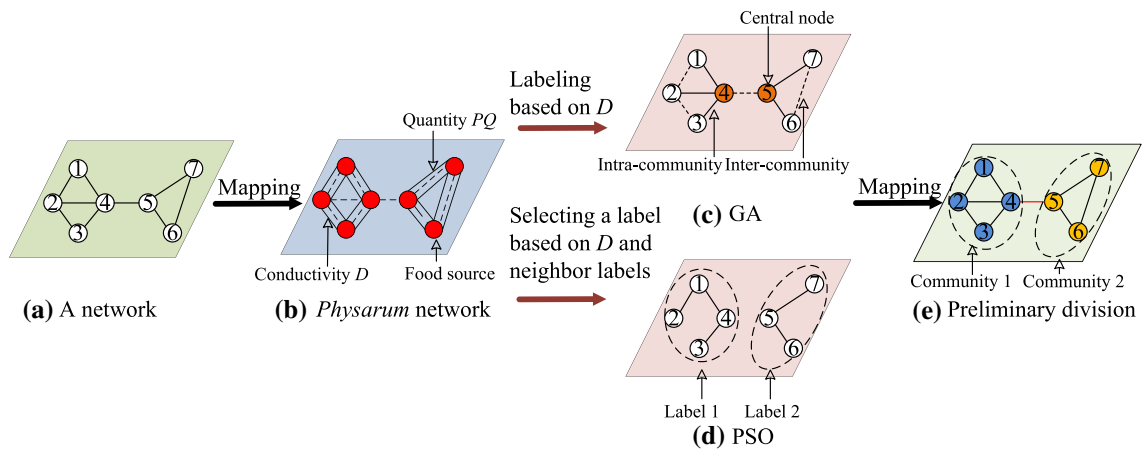


Fig. 1 The mapping of the PNM parameters to GA-based algorithms and PSO-based algorithms

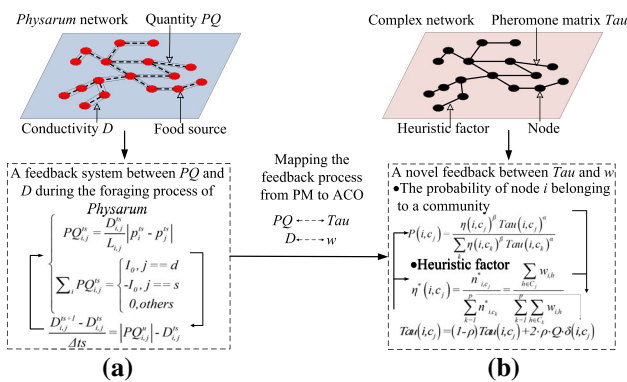


Fig. 2 The mapping of PNM parameters to ACO-based algorithms. **a** and **b** Compare the resemblance and distinctions between PNM- and ACO-based algorithms, respectively

In PSO algorithms, a particle randomly selects a neighbor label which dose not own the top 20% conductivities as its label. In Fig. 1d, n_1, n_2, n_3 and n_4 belong to community 1 and n_5, n_6 and n_7 are in community 2. Based on these methods, PNM is used to generate a preliminary community division, and then improve the quality of initial population. The final division will be the output as shown in Fig. 1e.

The mapping of PNM parameters to ACO-based algorithm is shown in Fig. 2. The resemblance and distinctions between PNM and ACO-based algorithms are explained in Figs. 2a and b. The conductivity D and the quantity of pipes in PNM are mapped to the pheromone matrix Tau and the heuristic factor η in ACO-based algorithms, respectively. In fact, there exists a positive feedback process both in PNM and ACO-based algorithms. The heuristic factor based on the conductivity of PNM induces each ant to produce a good solution, and the obtained community structures are measured by modularity. Then the solutions are fed back to pheromone matrix according to the pheromone update strategy, so that ants can communicate with each other better.

4 Experiments

4.1 Datasets

The experiments adopt six real networks and synthetic networks varying in sizes and scopes. The former is gathered by Newman¹, Batagelj and Mrvar², the latter is proposed by Lancichinetti [30]. Table 1 demonstrates the structure characters in networks. Specially, G_1 to G_4 are real-world and classic datasets with known community structures.

The details of parameters are shown in Tables 2, 3 and 4, which are based on the discussion of GA [12] [21], PSO [13] and ACO [14] [15], respectively. More specifically, the *adjacent rate* and *children proportion* are used to denote the initial proportion of centralized nodes and the proportion of new children for the next generation.

To distinguish the optimized algorithms from the original ones, a prefix (i.e., P-) is written in front of the original names of optimized algorithms. All experiments are operated over 100 runnings on average about the GA-based, PSO-based and ACO-based community discovery approaches.

4.2 Accuracy comparison

4.2.1 The comparison results of the modularity Q

Figures 3, 4, 5 and 6 first compare the Q values of original EA-based community discovery approaches and their enhanced approaches (i.e., with the prefix “P-”) based on our proposed *Physarum*-inspired computational framework. The ends of whiskers stand for the minimal and maximal Q values. Since the maximum and minimum are random, the contrast of those methods is centered on the evenness, first and third quartiles. There is a conclusion derived from the diagrams that

¹ <http://www-personal.umich.edu/~mejn/netdata/>.

² <http://vlado.fmf.uni-lj.si/vlado/vladonet.htm>.

Table 1 The essential characteristics of networks. G_i stands for the symbols of those networks. E and N indicate the total number of edges and nodes. C and k denote the clustering coefficient and averaged degree, severally. $\#C$ shows the size of communities in ground truth, where “-” signifies that the ground truth is unknown

G_i	Name	N	E	k	C	$\#C$
G_1	Karate club	34	78	4.588	0.588	2
G_2	Dolphins	62	160	5.129	0.303	2
G_3	PolBooks	105	441	8.400	0.488	3
G_4	Football	115	613	10.660	0.403	12
G_5	Jazz	198	2742	27.697	0.053	-
G_6	Celegans	297	1540	9.656	0.326	-
G_7	Synthetic network	100	479	9.58	0.141	3
G_8	Synthetic network	200	1384	13.84	0.080	2
G_9	Synthetic network	300	1619	10.79	0.376	5
G_{10}	Synthetic network	400	1300	6.50	0.484	3
G_{11}	Synthetic network	500	3966	15.86	0.053	6

Table 2 Parameters used in our GA-based experiments

Parameters	Values
Population size	200
Mutation rate	0.01
Children proportion	0.5
Adjacent rate	0.1
The maximum iterations	100

Table 3 Parameters used in our PSO-based experiments

Parameters	Values
Population size	100
Mutation rate	0.1
Inertia weight	0.7298
Learning factor	1.4961
The maximum iterations	100

the average Q values of three optimized algorithms in six real-world networks are higher than those of their original algorithms in most cases. And the 1st and 3rd quartiles of the P-approaches are higher than those of original approaches. It means that the enhanced algorithms based on PNM framework possess a better global exploration ability. The diagrams show that the optimized algorithms, P-ACOC and P-IACONet, get slightly lower medians of Q than the original algorithms in G_5 . However, the medians and averages of Q returned by the improved approaches are almost higher than those of primary approaches in all networks. What is more, the difference between the first and the third quartiles of Q reflects the fluctuation of Q . In rare cases, the algorithms could attain a wider range of Q values after optimization. For P-NGACD, its Q values vary as narrowly as NGACD's in G_2 . The same phenomena happen to both P-ACOC and P-IACONet in G_6 . However, in most cases, the Q values of the optimized methods vary in a smaller range.

Table 4 Parameters used in our ACO-based experiments

Parameters	ACOCIACONet	
Ants population	15	15
Evaporation rate ρ	0.1	0.2
The importance of pheromone trail factor α	1	1
The importance of heuristic factor β	2	2
Mutation rate	0.05	0.2
The maximum iterations	1000	50

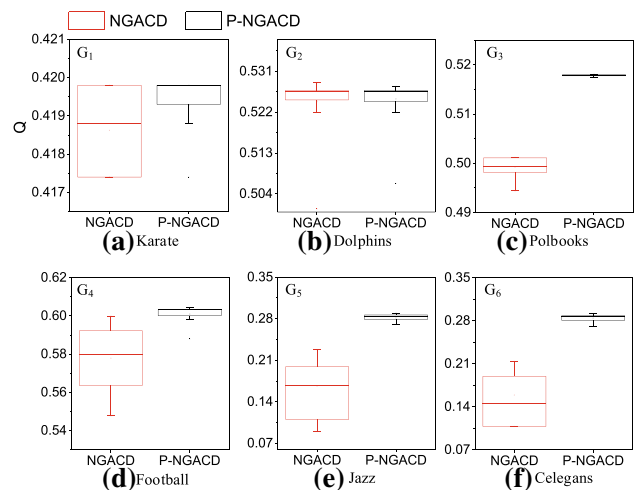


Fig. 3 The values of Q in box charts for P-NGACD and NGACD. The first and third quartiles are represented by the bottom and top of the box. The median is the band inside the box

Table 5 reports the averaged Q values in five synthetic networks, as listed in Table 1, which also verifies the optimization performance of our proposed framework. P-GDPSO can attain the highest Q values in G_7 , G_9 , G_{10} and G_{11} , and P-NGACD performs best in G_8 . In conclusion, the PNM framework improves the global search ability of original approaches in synthetic networks in diverse scopes.

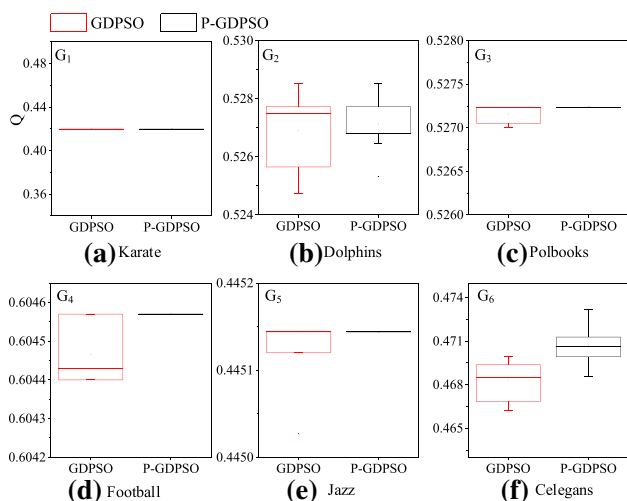


Fig. 4 The Q values in box charts for GDPSO and its optimized algorithms P-GDPSO in different kinds of networks

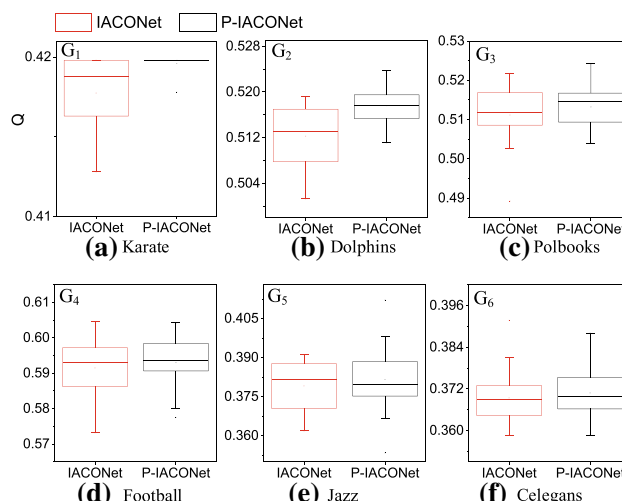


Fig. 6 The Q values in box charts for IACONet and its optimized algorithms P-IACONet in different kinds of networks. Compared to Fig. 5, we find that P-IACONet can achieve higher Q values than P-ACOC which is caused by the inherent performance difference between IACONet and ACOC

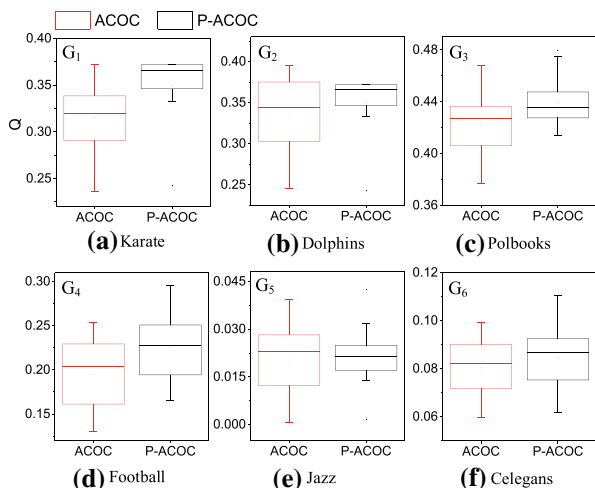


Fig. 5 The Q values in box charts for ACOC and its optimized algorithms P-ACOC in different kinds of networks

Furthermore, some optimization-based, heuristic-based community detection and network representation learning algorithms are employed to evaluate the efficiency of our approach. Specifically, there are evolutionary algorithms (i.e., ACO [32] and GA-Net [33]), swarm intelligence method (i.e., RWACO [34] and D-ACOC [38]), clustering method based on betweenness (i.e., GN [7]), label propagation based algorithm (i.e., LPA [36]), Markov clustering approach (i.e., MCL [37]) and network representation learning algorithms (i.e., Node2vec, DeepWalk, M-NMF, Walklets and GEM-SEC). Table 6 reports the modularity Q values returned by those algorithms in six small networks. P-GDPSO can achieve better results than others, which has the highest modularity Q values from G_1 to G_6 . In summary, optimized algorithms based on the PNM framework exhibit better search ability.

Table 5 The comparison of the modularity Q in synthetic networks

Alg.	Net.				
	G_7	G_8	G_9	G_{10}	G_{11}
GA-Net [33]	0.0407	0.0014	0.3364	0.5297	0.0080
NGACD [12]	0.2300	0.2000	0.2548	0.2972	0.2100
P-NGACD	0.2590	0.2491	0.2555	0.3157	0.2498
GDPSO [13]	0.3613	0.2442	0.5012	0.6534	0.4225
P-GDPSO	0.3614	0.2463	0.5017	0.6615	0.4226
ACO [32]	0.2310	0.1035	0.2599	0.3034	0.0923
IACONet [14]	0.3229	0.1749	0.3956	0.6163	0.2217
P-IACONet	0.3249	0.1749	0.3999	0.6179	0.2233

The bold numbers denote the best results

Table 6 The comparison with other approaches in the aspect of Q

Alg.	Net.					
	G_1	G_2	G_3	G_4	G_5	G_6
NGACD [12]	0.4186	0.5243	0.4982	0.5777	0.2278	0.1896
P-NGACD	0.4193	0.5249	0.5180	0.6008	0.2896	0.2891
IACONet [14]	0.4175	0.5153	0.5136	0.5914	0.3758	0.3804
P-IACONet	0.4197	0.5168	0.5204	0.5930	0.4066	0.3846
GDPSO [13]	0.4198	0.5278	0.5271	0.6045	0.4450	0.4683
P-GDPSO	0.4198	0.5284	0.5272	0.6046	0.4451	0.4693
GN [7]	0.4013	0.4706	0.5168	0.5996	0.4051	0.2280
MCL [37]	0.3715	0.5039	0.5199	0.5191	0.2653	0.3794
LPA [36]	0.3705	0.4806	0.5042	0.5884	0.2820	0.2318
RWACO [34]	0.3715	0.3774	0.4569	0.6010	0.2849	0
ACO [32]	0.4190	0.4930	0.4760	0.5310	0.2450	0.2529
GA-Net [33]	0.4060	0.4670	0.4900	0.5980	0.2652	0.2899
D-ACOC [38]	0.3718	0.4855	0.4955	0.4811	0.2733	0.2743
Node2vec [39]	0.3694	0.4002	0.4991	0.6006	0.2676	0.3750
DeepWalk [40]	0.3700	0.4002	0.4986	0.5954	0.2657	0.3809
M-NMF [41]	0.1559	0	0.5204	0.4585	0.2585	0.0136
Walklets [42]	0.3700	0.4002	0.5017	0.6010	0.2935	0.3812
GEMSEC [43]	0.3823	0.4685	0.4996	0.5418	0.2755	0.3280

The bold numbers denote the best results

4.2.2 The comparison results of convergence rate

This section estimates the convergence rate of the optimized algorithms. First, taking six real-world networks as examples, Fig. 7 plots the dynamical variations of Q during the iteration process. The Q is a criterion for estimating network partition returned by approaches. The greater the value of Q is, the higher the efficiency of an algorithm is. As demonstrated in Figs. 7a and b, there is no obvious difference between NGACD and P-NGACD. It means that our framework slightly improve the efficiency of NGACD in G_1 and G_2 . However, from Figs. 7c–f, the initial Q of P-NGACD is demonstrated higher when compared with NGACD. Simultaneously, the ability of searching the community structure strengthens and the values of Q increase with the generation process. As a result, the chart shows that the Q values of P-NGACD are higher than those of NGACD at each generation. The growth of generation steps narrows the gap between them. However, P-NGACD performs better because of the faster rate of convergence and the fewer iteration steps for optimal results. In particular, at the 50th generation, P-NGACD converges while NGACD does otherwise, which indicates that the improved strategy help algorithms have greater convergence. P-NGACD has an advantage over calculating Q on the entire iteration procedure.

After that, Fig. 8 depicts the iteration curve of the averaged Q values in six networks about the P-GDPSO and GDPSO algorithms. The results show that P-GDPSO is superior to GDPSO in the whole datasets.

Finally, Figs. 9 and 10 report the dynamic changes of the averaged Q values about the improved and original algorithms in ACOC and IACONet. Results show that two kinds of optimized ACO-based community detection approaches grow more quickly compared with pri-

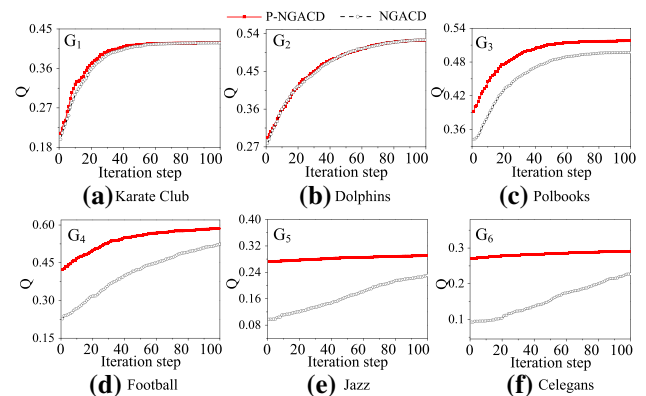


Fig. 7 The tendency of Q values with the increase of iterations in six real-world networks. The higher the Q value is, the better the division of the algorithm is

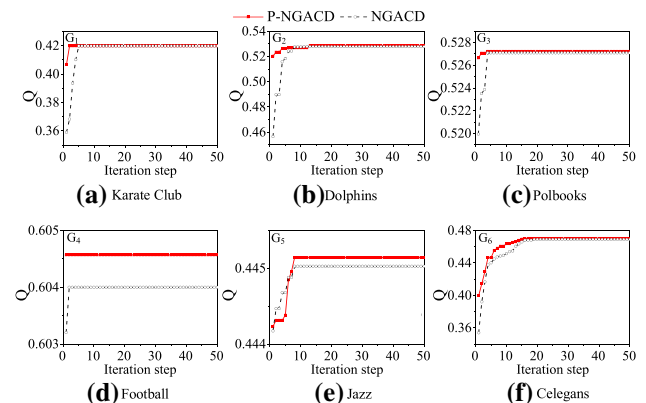


Fig. 8 The modularity of P-GDPSO is slightly better than that of the GDPSO in six real-world networks. Obviously, the improved algorithm expedites the rate of convergence

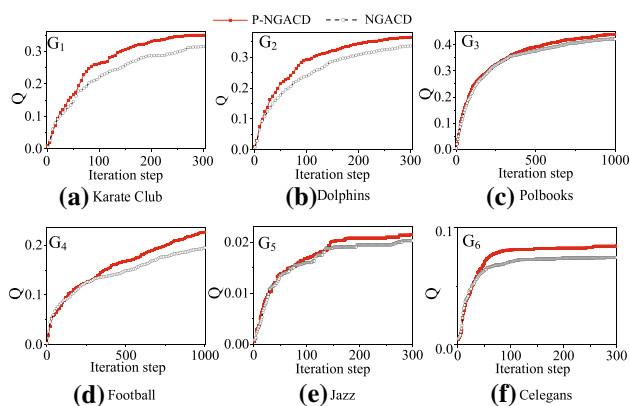


Fig. 9 The trend charts of the mean Q values of ACO and P-ACO with the enhancement of iteration in six real networks

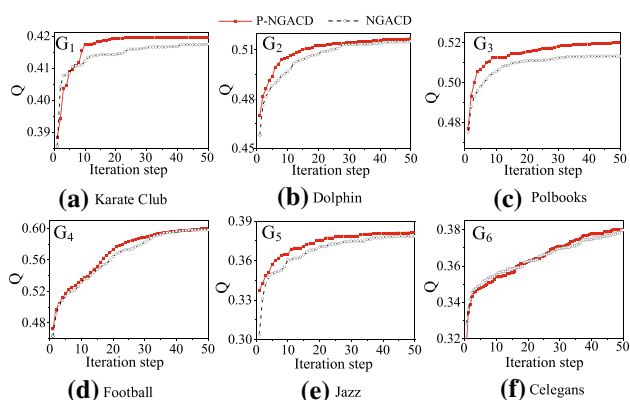


Fig. 10 The dynamical variations of the mean Q values of IACONet and P-IACONet with the enhancement of iteration in six real networks

mary approaches. As demonstrated in such figures, the initial Q values of optimized approaches approximate those of the original approaches. Nonetheless, optimized approaches higher increase, in comparison with the original ACO approaches. With the increment of iterations, the difference between original and optimized approaches emerges. These phenomena are caused by the positive feedback system. The optimized ACO algorithms generate good initial solutions and obtain divisions with a high Q value. Then results are fed back to the pheromone matrix by an updating strategy with respect to Q .

4.2.3 The visualization results for real-world networks

According to the visualization results, colors denote the partitions of networks returned by algorithms and geometric figures display the real partitions. Taking GA-based algorithms as an example, Figs. 11a and b illustrates the visualization results in four small real-world networks. The circle and rectangle indicate diverse real clubs.

Figures 11c and d show the division returned by P-NGACD and NGACD for the dolphins network. Based

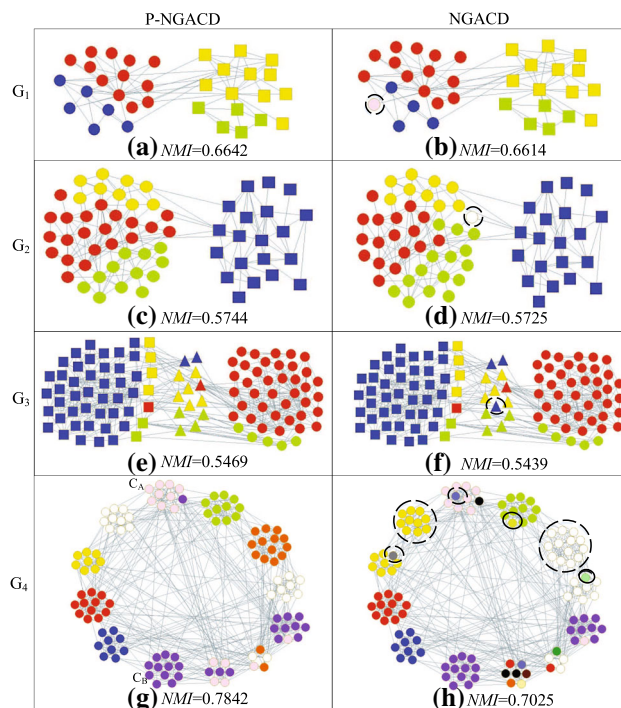


Fig. 11 Partition results of NGACD and P-NGACD on the basis of the top value of objective function Q in (a)–(h). More specifically, the averaged NMI values over 100 repeated runnings are reported below each subgraph. The shapes of the nodes stand for the ground truth, and the colors represent community partitions of approaches. The dotted circle highlights the major diversity in partitions

on the benchmark criteria, rectangles and circles stand for real communities. As shown in Fig. 11c, although the communities detected by P-NGACD do not stay the same as the real communities, P-NGACD distributes a big community into three small ones. Figures 11e and f are the partition results of two algorithms in the Polbooks network. Two algorithms produce the division denoted by four colors in the diagrams. Edges depict the references between them. There are four colors depicted in these figures which denote partition results obtained by two algorithms. The context of books bridges the distance of some books to become smaller communities. Although P-NGACD cannot detect real community represented by triangle well, most of books represented by circle and rectangle are divided into two communities successfully. Figures 11g and h show the community structures searched by two algorithms in the Football network. Each node stands for a football team, and the edges denote the matches between the two teams. The two marked communities C_A and C_B show the details of partitions. Although the purple node in C_A should not be involved in C_B according to the real division, the team represented by this node does have more games with the teams in C_B . In other words, those partitions returned by algorithms are better than the real ones from the aspect of edge density of commu-

Table 7 The comparisons in the aspects of *FVIC* and *dc*

Alg.	Net.			
	G_1	G_2	G_3	G_4
	<i>FVIC</i> (<i>dc</i>)	<i>FVIC</i> (<i>dc</i>)	<i>FVIC</i> (<i>dc</i>)	<i>FVIC</i> (<i>dc</i>)
NGACD [12]	0.9926 (4.1)	0.9863 (4.4)	0.8500 (6.7)	0.5535 (9.4)
P-NGACD	0.9912 (4.0)	0.9871 (4.3)	0.8576 (4.7)	0.6470 (7.8)
GDPSO [13]	1.0000 (4.0)	0.9839 (4.9)	0.8571 (5.0)	0.8696 (10)
P-GDPSO	1.0000 (4.0)	1.0000 (4.0)	0.8571 (5.0)	0.8696 (10)
IACONet [14]	0.9902 (4.0)	0.9801 (5.2)	0.8469 (5.5)	0.6289 (10.2)
P-IACONet	1.0000 (4.0)	0.9855 (4.4)	0.8486 (4.6)	0.8337 (7.4)

Table 8 The basic features of a synthetic networks with different number of communities. *N* and *E* represent the total size of nodes and edges. *k* and *C* represent the averaged degree and clustering coefficient, severally. #*C* show the size of communities in the ground truth

G_{12}	<i>N</i>	<i>E</i>	<i>k</i>	<i>C</i>	# <i>C</i>
G_{12}^1	500	4921	19.684	0.157	5
G_{12}^2	500	4808	19.232	0.182	6
G_{12}^3	500	4851	19.404	0.212	7
G_{12}^4	500	5036	20.144	0.263	8
G_{12}^5	500	4966	19.864	0.280	9
G_{12}^6	500	4829	19.316	0.310	10

nities. From illustrations above, we can conclude that P-NGACD behaves better in community detection.

Besides *NMI* reported in Fig. 11, Table 7 shows the criterion of *FVIC* and the number of detected clusters (*dc*) in different synthetic networks with known divisions. Results show that our proposed optimization framework can reduce the unreasonable fragments by measuring the values of *dc* and improve the identified accuracy of each node.

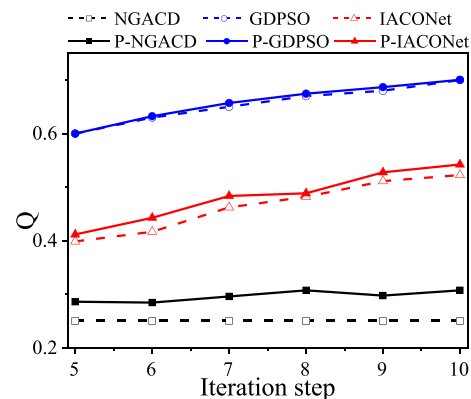


Fig. 12 The dynamical variation of averaged *Q* in synthetic networks G_{12} with different number of communities

4.3 Discussion

4.3.1 Scalability analysis

To further assess the efficiency of our proposed method, some synthetic networks with different number of communities are first constructed [30]. The features of these networks are listed in Table 8. Then we analyze the effect of the number of communities on the efficiency of our method.

As shown in Fig. 12, optimized algorithms always outperform original algorithms in terms of *Q* values. We can conclude that our proposed framework can optimize original algorithms independently of the community number. More specifically, the GA-based algorithm is stable with the increment of community number. However, the performances of PSO-based and ACO-based algorithms keep increasing with the increment of community number.

4.3.2 Statistical analysis

According to the statistical method in [19], we implement some statistical analyses (i.e., Friedman test, Bonferroni–Dunn’s test, and Holm’s and Hochberg’s methods) for comparing performance between our proposed method and other methods. To begin with, Friedman test (i.e., χ^2_F) in Table 9, is used to rank the differences of algorithms in Tables 5 and 6. Based on Eq. (19), the χ^2_F of Table 9 is 53.0932. Due to $\chi^2_F > \chi^2_{0.05} = 14.07$ in the Chi-square table, there is a conclusion that results of algorithms are quite different with a confidence of 95%. Then this paper implements the Bonferroni–Dunn’s test to measure the specific

Table 9 The rankings obtained based on Tables 5 and 6. The value is the ranking result of approaches in every dataset

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	R
P-GDPSO	1.5	1	1	1	1	1	1	1	1	1	1	1.046
GDPSO	1.5	2	2	2	2	2	2	2	2	2	2	1.955
IACONet	5	4	3	4	3	3	3	4	3	3	3	3.455
NGACD	4	3	4	5	6	6	5	3	6	6	4	4.727
ACO	3	5	6	6	5	5	4	5	5	5	5	4.909
GA-Net	6	6	5	3	4	4	6	6	4	4	6	4.909

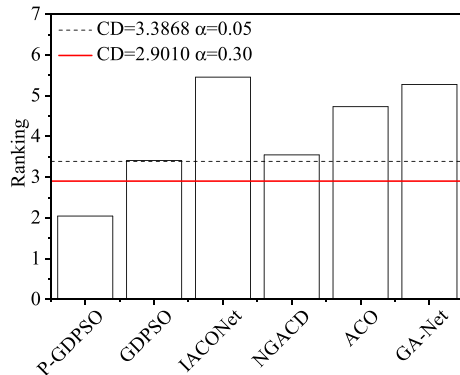


Fig. 13 The Bonferroni–Dunn’s graph corresponding to the results of Table 9. The horizontal line denotes the value which equals to the sum of the ranking of the control algorithm (i.e., P-GDPSO) and the corresponding CD . Those bars which exceed this line are associated to an algorithm with worse performance than P-GDPSO

diversities between two approaches. Based on Eq. (20), we obtain the critical difference (i.e., CD_α) with diverse confidence levels in Table 9, such as $CD_{0.05} = 2.3413$ and $CD_{0.3} = 1.8555$. Therefore, the conclusion can be acquired that P-GDPSO is better than five approaches (i.e., GDPSO, IACONet, NGACD, ACO, GA-Net) with 70% confidence (i.e., $\alpha=0.3$) and 95% confidence (i.e., $\alpha=0.05$) as shown in Fig. 13.

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right], \quad (19)$$

where N and k denote the number of datasets and algorithms, respectively. R is the Ranking R in Table 9.

Finally, this paper applies the Holm’s and Hochberg’s methods to further compare the diversities between two approaches. Based on Eq. (21), the statistic for competing the method i with the method j (represented as z value) is calculated. To be specific, the value of Unadjusted p is obtained by seeking for the standard distribution table on the basis of z value. And Bonferroni–Dunn p is computed by $BD_{p_i} = \min\{v_i; 1\}$, where $v_i = (k-1)U_{p_i}$. Holm p is computed by $H_{p_i} = \min\{v_i; 1\}$, where $v_i = \min\{(k-j)U_{p_j} : 1 \leq j \leq i\}$. Hochberg p is computed by $HB_{p_i} = \min\{(k-j)U_{p_j} : (k-1) \geq j \geq i\}$. According to such comparison, we can provide more detailed information to conclude whether a control algorithm is better than others. Table 10 reports the statistical results of Table 9. There is a conclusion that P-GDPSO outperforms five approaches (i.e., GDPSO, IACONet, NGACD, ACO, GA-Net) with 95% confidence (i.e., $\alpha = 0.05$) and 80% confidence (i.e., $\alpha = 0.2$).

$$CD_\alpha = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \quad (20)$$

where α is a confidence level and q_α can be obtained in the critical value of Z table. k and N represent the number of datasets and algorithms, respectively.

$$z = \frac{R_i - R_j}{\sqrt{\frac{k(k+1)}{6N}}}, \quad (21)$$

where R denotes the Ranking R in Table 9. N and k are the number of datasets and algorithms, respectively.

Table 10 The p value on $G_1 - G_{11}$ (P-GDPSO is the control method)

P-GDPSO vs.	z	Unadjusted p	Bonf.-Du. p	Holm p	Hochberg p
GA-Net	4.84	0.0000004	0.0000002	0.0000002	0.0000020
ACO	4.84	0.0000004	0.0000002	0.0000002	0.0000016
NGACD	4.62	0.0000100	0.0000050	0.0000030	0.0000300
IACONet	3.02	0.0025200	0.012600	0.015120	0.0151200
GDPSO	1.14	0.2542000	1.000000	0.254200	0.2542000

5 Conclusion

Some studies have been carried out to apply the optimization-based algorithms for community detection. Nonetheless, because of the inherent complexity of discovering community structure, the computational efficiency should be further improved. We have two major contributions: (1) a modified *Physarum* network model (PNM) for community mining is introduced, which can distinguish edges in different communities from those in the same community; (2) our proposed method takes the advantage of valuable prior knowledge into consideration, which has an excellent enhancement in the robustness and accuracy, compared with other optimization-based algorithms; and (3) our framework is extensible, which has confirmed by a computational complexity analysis.

This work was supported by the National Key R&D Program of China (No. 2020AAA0107700), National Natural Science Foundation of China (Nos. 61976181, 11931015, U1803263), Fok Ying-Tong Education Foundation, China (No. 171105), Key Technology Research and Development Program of Science and Technology Scientific and Technological Innovation Team of Shaanxi Province (No. 2020TD-013), the Science and Technology Foundation of Guizhou (No. QKHJC20181083) and the Science and Technology Support Program of Guizhou (No. QKHZC2021YB531).

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: All network used in Sec. 4.1 are public datasets and cited in terms of footnote in Page 6].

References

- C. Gao, J.M. Liu, IEEE Trans. Syst. Man Cybern. -Syst. **47**, 171 (2017)
- S.D. Li, L.Y. Jiang, X.B. Wu et al., Appl. Math. Comput. **401**, 126012 (2021)
- P. Zhu, X. Dai, X. Li et al., Europhys. Lett. **126**, 48001 (2019)
- B. Luzar, Z. Levnajic, J. Povh et al., PLoS One **9**, e94429 (2014)
- Z. Wang, C.Y. Wang, C. Gao et al., Sci. China-Inf. Sci. **63**, 212205 (2020)
- C. Gao, Z. Su, J.M. Liu, J. Kurths, Commun. ACM **62**, 61–67 (2019)
- M. Girvan, M.E.J. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002)
- H.J. Li, L. Wang, Y. Zhang et al., New J. Phys. **22**, 063035 (2020)
- M.G. Gong, Q. Cai, X. Chen et al., IEEE Trans. Evol. Comput. **18**, 82 (2014)
- Z. Wang, C.Y. Wang, X.H. Li et al., IEEE Trans. Knowl. Data Eng. (2020). <https://doi.org/10.1109/TKDE.2020.2997043>
- C.Y. Wang, F. Zhang, Y. Deng et al., Chaos Soliton Fract **138**, 109886 (2020)
- X.H. Li, C. Gao, P.R. Yang, in *10th International Conference on Natural Computation (ICNC), Xiamen, 2014*, p. 486. <https://ieeexplore.ieee.org/document/6975883>
- Q. Cai, M.G. Gong, L.J. Ma et al., Inf. Sci. **316**, 503 (2015)
- C. Mu, J. Zhang, L. Jiao, in *Proceedings of IEEE Congress on Evolutionary Computation, Beijing, 2014*, p. 700. <https://ieeexplore.ieee.org/document/6900411>
- P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, Anal. Chim. Acta **509**, 187 (2004)
- C. Gao, M.X. Liang, X.H. Li et al., IEEE/ACM Trans. Comput. Biol. Bioinf. **15**, 1916 (2018)
- A. Tero, S. Takagi, T. Saigusa et al., Science **327**, 439 (2010)
- C. Gao, C. Liu, D. Schenz et al., Phys. Life Rev. **29**, 1 (2019)
- C. Gao, S. Chen, X.H. Li et al., Appl. Soft. Comput. **61**, 239 (2017)
- Y.T. Lu, M.X. Liang, C. Gao et al., In *International Conference on Natural Computation (Fuzzy Systems and Knowledge Discovery, Changsha, 2016)*, p. 673
- M. Tasgin, A. Herdagdelen, H. Bingol, Comput. Res. Repos. **2005**, 1067 (2007)
- D. Jin, D.X. He, D.Y. Liu, et al., in *22nd International Conference on Tools with Artificial Intelligence, Arras, 2010*, p. 105. <https://ieeexplore.ieee.org/document/5670026/>
- M. Dorigo, M. Birattari, T. Stutzle, I.E.E.E. Comput. Intell. Mag. **1**, 28 (2006)
- D. Jin, D.Y. Liu, B. Yang et al., Adv. Complex Syst. **14**, 795 (2011)
- X.H. Li, Z.X. Wang, T.Y. Lu et al., J. Bionic Eng. **6**, 77 (2009)
- A. Tero, R. Kobayashi, T. Nakagaki, J. Theor. Biol. **244**, 553 (2007)
- C. Gao, Y. Fan, S.H. Jiang et al., IEEE Trans. Intell. Transp. Syst. (2021). <https://doi.org/10.1109/TITS.2021.3058185>
- M.X. Liang, C. Gao, Z.L. Zhang, Nat. Comput. **16**, 85 (2017)
- F. Wu, B.A. Huberman, Eur. Phys. J. B **38**, 331 (2004)
- A. Lancichinetti, S. Fortunato, F. Radicchi, Phys. Rev. E **78**, 046110 (2008)
- A. Clauset, C.R. Shalizi, M.E.J. Newman, SIAM Rev. **51**, 661 (2009)
- H.H. Chang, Z.R. Feng, Z.G. Ren, in *IEEE Congress on Evolutionary Computation, Cancun, 2013*, p. 3072. <https://ieeexplore.ieee.org/document/6557944>
- C. Pizzuti, in *Proceedings of International Conference on Parallel Problem Solving from Nature, Dortmund, 2008*, p. 1081. https://doi.org/10.1007/978-3-540-87700-4_107
- D. Jin, B. Yang, J. Liu et al., J. Softw. **23**, 451 (2012)
- M.E.J. Newman, Phys. Rev. E **69**, 066133 (2004)
- U.N. Raghavan, R. Albert, S. Kumara, Phys. Rev. E **76**, 036106 (2007)
- V. Satuluri, S. Parthasarath, in *Proceedings of the International Conference on Knowledge Discovery and Data Mining, Paris, 2009*, p. 737
- J.J. Chen, S.P. Gao, Z. Su et al., In *Proceedings of the International Conference on Natural Computation, Fuzzy Systems and Knowledge Discov-*

- ery, Kunming, 2019, p. 141. https://doi.org/10.1007/978-3-030-32456-8_15
39. A. Grover, J. Leskovec, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, 2016*, p. 855. <https://doi.org/10.1145/2939672.2939754>
40. B. Perozzi, R. Al-Rfou, S. Skiena, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, 2014*, p. 701. <https://doi.org/10.1145/2623330.2623732>
41. X. Wang, P. Cui, J. Wang, et al., in *Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, 2017*, p. 203. <https://doi.org/10.5555/3298239.3298270>
42. B. Perozzi, V. Kulkarni, H. Chen, et al., in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Sydney, 2017*, p. 258. <https://doi.org/10.1145/3110025.3110086>
43. B. Rozemberczki, R. Davies, R. Sarkar, et al., in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Vancouver, 2019*, p. 65. <https://doi.org/10.1145/3341161.3342890>