Multi-objective ant colony optimization algorithm based on decomposition for community detection in complex networks

Caihong Mu[a], Jian Zhang[a], Yi Liu[b,*], Rong Qu[c], Tianhuan Huang[a]

*Corresponding author:

Email: yiliuxd@foxmail.com

[a] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an, 710071, China

[b]School of Electronic Engineering, Xidian University, Xi'an, 710071, China

[c]School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK

Abstract

Community detection aims to identify topological structures and discover patterns in complex networks, which presents an important problem of great significance. The problem can be modeled as an NP hard combinatorial optimization problem, to which multi-objective optimization has been applied, addressing the common resolution limitation problem in modularity-based optimization. In the literature, ant colony optimization (ACO) algorithm, however, has been only applied to community detection with single objective. This is due to the main difficulties in defining and updating the pheromone matrices, constructing the transition probability model, and tuning the parameters. To address these issues, a multi-objective ACO algorithm based on decomposition (MOACO/D-Net) is proposed in this paper, minimizing negative ratio association and ratio cut simultaneously in community detection. MOACO/D-Net decomposes the community detection multi-objective optimization problem into several subproblems, and each one corresponds to one ant in the ant colony. Furthermore, the ant colony is partitioned into groups, and ants in the same group share a common pheromone matrix with information learned from high quality solutions. The pheromone matrix of each group is updated based on updated nondominated solutions in this group. New solutions are constructed by the ants in each group using a proposed transition probability model and each of them is then improved by an improvement operator based on the definition of strong community. After improvement, all the solutions are compared with the solutions in the external archive and the nondominated ones are added to the external archive. Finally each ant updates its current solution based on a better neighbor, which may belong to an adjacent group. The resulting final external archive consists of nondominated solutions, and each one corresponds to a different partition of the network. Systematic experiments on LFR benchmark networks and eight real-world networks demonstrate the effectiveness and robustness of the proposed algorithm. The ranges of proper values for each parameter are also analyzed, addressing the key issue of parameter tuning in ACO algorithms based on a large number of tests conducted.

Keywords

Complex networks; Community detection; Multi-objective optimization; Ant colony optimization

1 Introduction

Many real-world systems can be regarded as complex networks, such as interpersonal relationship networks and scientific collaboration networks in social systems, protein networks and neuron networks in ecosystems, and internet in science and technology systems. Detecting community structures and analyzing functions in the complex network presents one of the most important and significant interdisciplinary research challenges to current scientific communities (Fortunato and Hric 2016; Guédon and Vershynin 2016; Zhang and Zhou 2016; Lyzinski et al. 2017; Schaub et al. 2017; Zhou et al. 2017).

Mathematically complex networks can be modeled as graphs, where vertices and edges in a network can be modeled as nodes and links in a graph respectively (Fortunato 2010; Newman 2011). A community or a cluster is defined as a set of vertices whose connections should be dense inside and sparse with other outside communities (Radicchi et al. 2004; Newman and Girvan 2004).

In the recent optimization-based algorithms applied to community detection, modularity of networks introduced by Newman and Girvan (2004) has been the most widely used objective function. It was originally applied as a stopping criterion in the algorithm, and quickly became an essential measure in many clustering methods. Newman (2004) devised a greedy method named Fast Newman (FN) algorithm to optimize the modularity of networks for the first time. It is an agglomerative hierarchical clustering method where vertices are merged to form larger communities if the modularity of the network increases. More modularity-based optimization algorithms, such as spectral optimization (Newman 2006) and evolutionary algorithms (EAs), have been later proposed to solve community detection problems. CNM algorithm proposed by Clauset, Newman and Moore (2004) is the improvement version of FN algorithm. By exploiting some shortcuts in the optimization problem and using more sophisticated data structures, CNM runs far more quickly and can deal with large scale network. Based on CNM, Mu et al. (2014) proposed a two-stage algorithm called CNM-IC with an influence coefficient, which can detect the hierarchical, non-overlapping and overlapping community structure.

There is a resolution limitation problem, however, in modularity-based algorithms (Fortunato, and Barthélemy, 2007). It means that modularity optimization may fail to identify communities smaller than a scale, depending on the overall size of the network and the degree of interconnectedness between the communities. A new measure called modularity density was thus proposed by Li et al. (2008) to address this issue. The experimental results showed that, using tunable coefficient of different values, modularity density can reveal communities at different hierarchical levels. Gong et al. (2012) solved community detection as a multi-objective optimization problem by using the multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang et al. 2007), where modularity density was divided into two parts: ratio association (Angelini et al. 2007) and ratio cut (Wei et al. 1991). Therefore different community structures can be revealed in a single run.

Besides EAs, ant colony optimization (ACO) algorithm is also a member of bionic algorithms. ACO was firstly proposed by Dorigo (1992), inspired by the behavior of foraging of ant colonies. It is a population-based and parallel algorithm with outstanding global search ability, and has been integrated with other algorithms (Ji et al. 2011). ACO has been successfully applied to many NP hard combinatorial optimization problems including traveling salesman problem (Dorigo and Gambardella 1997), job shop scheduling (Colorni et al. 1994), graph coloring (Costa and Hertz 1997) and vehicle routing (Bullnheimer et al. 1999). An algorithmic framework for continuous ACO algorithms called UACOR, from which earlier continuous ACO algorithms can be instantiated, was introduced by Liao et al. (2014). Ke and Zhang et al. (2013) introduced a multi-objective ACO algorithm called MOEA/D-ACO which combines ACO and MOEA/D. MOEA/D-ACO showed good performance in solving the multi-objective traveling salesman problem and multi-objective 0-1 knapsack problem.

In recent years ACO has been applied to the community detection problem. He et al. (2011) proposed a Multi-layer ACO without employing the pheromone parameter to reduce the tuning time. Each ant in the algorithm only decides whether its current vertex belongs to the community of its preceding vertex. To further improve the algorithm performance, a Markov random walk theory-based ACO was proposed by Jin et al. (2011), where each ant detects its own community by using a new random walk model. Chang et al. (2013) suggested a modularity-based optimization method named ACO-Net which updates pheromone trails within the Max-Min Ant System (MMAS) (Stützle and Hoos 2000). To avoid redundant computing and premature convergence in ACO-Net, Mu et al. (2014) introduced an intelligent ACO (IACO-Net), using intelligent ants with proactive-learning and self-learning to explore the search space in a more efficient and stable way.

Other issues in applying ACO-based algorithms to community detection are worth further investigation. Firstly results obtained by ACO are quite sensitive to parameter tuning on the pheromone mechanism and probabilistic choosing model. These include the pheromone information factor α, the heuristic information factor β, and the persistence ratio ρ. Secondly, like in the other single objective optimization algorithms, the resolution limitation problems also occur in ACO for detecting communities in complex networks. Many runs are needed to tune the parameters in single objective ACO to obtain partitions at different hierarchical levels.

A multi-objective ant colony optimization algorithm based on decomposition (MOACO/D-Net) is proposed in this paper to address the above mentioned issues and solve the community detection as a multi-objective optimization problem. The problem is divided into a number of subproblems, and each one is addressed by an ant in the ant colony and corresponds to a particular point in the Pareto Front (PF). To improve the communication between the ants, the ant colony is partitioned into several groups, and each has its neighbors which may belong to an adjacent group. In each generation, each ant constructs a new solution according to the proposed transition probability model. The ants in the same group share a common pheromone information matrix, which is updated based on the information extracted from the newly constructed solutions with good quality in this group that are nondominated by the solutions in external archive and then added into it. At the end of each generation, each ant then updates its current solution by a better neighbor. In addition to systematic experiments on LFR benchmark networks (Lancichinetti et al. 2008) and eight real-world networks, the ability of MOACO/D-Net on detecting communities at different hierarchical levels on a real-world network is demonstrated.

The contributions of this paper are as follows. (1) A new solver named MOACO/D-Net is provided for the community detection problem by combining ACO and MOEA/D. (2) In MOACO/D-Net several new strategies are designed for solving community detection problem, including a new transition probability model and update of pheromone matrices based on the concepts of decomposition, as well as a new improvement operator based on the definition of strong community that produces better solutions with much less computation cost. (3) The key issue of parameter tuning in ACO algorithms is addressed by examining the ranges of proper values for each parameter in MOACO/D-Net based on abundant experiments conducted.

The rest of this paper is arranged as follows. Section 2 defines the community detection problem and introduces some basic concepts used in the decomposition-based multi-objective optimization algorithm. Section 3 presents the details of MOACO/D-Net. Section 4 presents the experiments and analysis. Finally conclusions are provided in Section 5.

2 Basic concepts

2.1 Multi-objective optimization

A multi-objective optimization problem can be defined (Gong et al. 2012) as follows:

$$\text{minimize } \boldsymbol{F}(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_m(\boldsymbol{x})\right)^T \quad \text{subject to } \boldsymbol{x} \in \Omega \tag{1}$$

where $\boldsymbol{x}$ is a decision vector, Ω is the decision space and $\boldsymbol{F}$ includes m objective functions. Assuming $\boldsymbol{x}_A, \boldsymbol{x}_B \in \Omega$ are two different decision vectors of a minimization problem, $\boldsymbol{x}_A$ dominates $\boldsymbol{x}_B$ (written as $\boldsymbol{x}_A \succ \boldsymbol{x}_B$) if:

$$\forall i = 1,2,\dots,m: f_i(\boldsymbol{x}_A) \le f_i(\boldsymbol{x}_B) \ \wedge \ \exists j = 1,2,\dots,m: f_j(\boldsymbol{x}_A) < f_j(\boldsymbol{x}_B) \tag{2}$$

$\boldsymbol{x}^* \in \Omega$ is a Pareto-optimal solution or nondominated solution if there exists no another $\boldsymbol{x} \in \Omega$ such that $\boldsymbol{x} \succ \boldsymbol{x}^*$. The set of all nondominated solutions is called the Pareto-optimal set which can be defined as:

$$P^* \triangleq \{\boldsymbol{x}^* \in \Omega | \neg\exists \boldsymbol{x} \in \Omega,\ \boldsymbol{x} \succ \boldsymbol{x}^*\} \tag{3}$$

The Pareto Front PF^* is a surface in the objective function space, which consists of all the corresponding objective vectors of Pareto-optimal solutions in P^*.

$$PF^* = \left\{\boldsymbol{F}(\boldsymbol{x}^*) = \left(f_1(\boldsymbol{x}^*), f_2(\boldsymbol{x}^*), \dots, f_m(\boldsymbol{x}^*)\right)^T \middle| \boldsymbol{x}^* \in P^*\right\} \tag{4}$$

A multi-objective optimization algorithm aims to find sufficient nondominated solutions to approximate the true Pareto-optimal front.

2.2 Definition of community detection

Generally a complex network can be modeled as a graph, which is represented by $G = (V, E)$, with $n = |V|$ vertices and $n_e = |E|$ edges. *V* is a set of vertices and *E* is a set of edges between two vertices in *G*. $\boldsymbol{A}$ is an $n \times n$ adjacent matrix of graph *G*, and element A_{ij} is 1 if vertex v_i and vertex v_j are connected, otherwise $A_{ij} = 0$. The degree of vertex v_i can be expressed as $k_i = \sum_{j \in V} A_{ij}$. In a final partition $P = \{V_1, V_2, \dots, V_C\}$ of *G*, each element V_i is a proper subset of *V*, where *C* is the number of communities.

As mentioned above, connections between vertices in the same community should be dense and those between different communities should be sparse. Based on this principle, Radicchi et al. (2004) defined a community in a strong sense and in a weak sense, as described below.

For simplicity, a subset $V_i \in V$ is written as *U*. *U* is a community in the strong sense if $k_{in}^i > k_{out}^i,\ \ \forall i \in U$, and *U* is a community in the weak sense if $\sum_{i \in U} k_{in}^i > \sum_{i \in U} k_{out}^i$, where k_{in}^i (internal degree) is the number of edges between v_i and other vertices in *U*, k_{out}^i (external degree) is the number of edges of vertex v_i connected to the rest of vertices in *V*. Therefore each vertex in a strong community *U* has a larger internal degree and in a weak community *U* the sum of all the internal degree should be larger than that of all the external degree. Furthermore a strong community is also a weak community, but not vice versa.

2.3 Objective functions

Multi-objective optimization algorithms have already been applied to solve the community detection problem. A multi-objective genetic algorithm called MOGA-Net was introduced to find hierarchical structures (Pizzuti 2009), with two different objectives: community score, as proposed in the GA-Net (Pizzuti 2008), and community fitness (Lancichinetti et al. 2009). However, optimizing the community fitness does not lead to good partitions. Gong et al. (2014) proposed a multi-objective discrete particle swarm optimization algorithm named MODPSO with two objectives, namely kernel k-means and ratio cut. These objectives can be extended to a signed version, which is suitable for the signed network.

Considering the resolution limitation problem in modularity, modularity density can be used as the objective function for community detection (Li et al. 2008). Given a partition $P = \{V_1, V_2, \dots, V_C\}$ of an undirected graph, *C* is the number of communities. For $i = 1,2,\dots,C$, the modularity density is defined in Eq. (5) as:

$$D = \sum_{i=1}^{C} \frac{L(V_i, V_i) - L(V_i, \bar{V}_i)}{|V_i|} = \sum_{i=1}^{C} \frac{L(V_i, V_i)}{|V_i|} - \sum_{i=1}^{C} \frac{L(V_i, \bar{V}_i)}{|V_i|} \tag{5}$$

where V_i is the set of vertices in the i^{th} community and $\bar{V}_i = V - V_i$, $L(V_1, V_2) = \sum_{i\in V_1, j\in V_2} A_{ij}$.

The larger the value D is, the more accurate a partition is. In fact, the first term of D is equivalent to the ratio association (Angelini et al. 2007), and the other term is equivalent to the ratio cut (Wei et al. 1991). The modularity density D can thus be seen as a combination of the ratio association and the ratio cut. Generally maximizing the ratio association often partitions a network into small communities with dense interconnections, while minimizing the ratio cut often partitions a network into large communities with sparse connections with the others.

In this paper, Negative Ratio Association (NRA) and Ratio Cut (RC) are employed in the objective function vector, and is minimized as shown in Eq. (6).

$$\text{minimize } \boldsymbol{F}(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x})\right)^T, \qquad \begin{cases} f_1(\boldsymbol{x})\text{: NRA} = -\sum_{i=1}^{C} \frac{L(V_i, V_i)}{|V_i|} \\ f_2(\boldsymbol{x})\text{: RC} = \sum_{i=1}^{C} \frac{L(V_i, \bar{V}_i)}{|V_i|} \end{cases} \tag{6}$$

2.4 Decomposition of multi-objective optimization problems

In the literature, several approaches have been used to decompose a multi-objective optimization problem into N subproblems (Miettinen 1999; Adriano and Paolo 1984; Ehrgott 2005; Eichfelder 2008). The two most commonly used approaches are weighted sum approach and Tchebycheff approach. In our work, the Tchebycheff approach (Zhang et al. 2007) is employed to convert the multi-objective community detection problem into a number of single-objective subproblems, each modeled as a minimization problem shown in Eq. (7).

$$g(\boldsymbol{x}|\boldsymbol{\lambda}) = \max_{1\le i\le m}\{\lambda_i |f_i(\boldsymbol{x}) - z_i^*|\} \qquad z_i^* = \min_{x\in\Omega} f_i(\boldsymbol{x}) \ \ (i = 1, \dots, m) \tag{7}$$

z_i^* is a reference point. In a maximization problem, $z_i^* = max_{x\in\Omega} f_i(\boldsymbol{x})$.

Let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ be a weight vector, where $\lambda_i \ge 0$ for $i = 1,2, \dots, m$ and $\sum_{i=1}^{m} \lambda_i = 1$. Setting of N and $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2 \dots, \boldsymbol{\lambda}^N$ is controlled by parameter H. More precisely, $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2 \dots, \boldsymbol{\lambda}^N$ are all the weight vectors and each individual weight component of them takes a value from

$$\left(\frac{0}{H}, \frac{1}{H}, \dots, \frac{H}{H}\right) \tag{8}$$

Therefore, the number of the weight vectors is $N = C_{H+m-1}^{m-1}$. In MOACO/D-Net, the number of objective functions m is 2, and N is the number of ants, so $H = N - 1$. Thus $\boldsymbol{\lambda}^j, j \in \{1,2, \dots, N\}$, is a vector with two components:

$$\boldsymbol{\lambda}^j = \left(\frac{j-1}{N-1}, \frac{N-j}{N-1}\right) \tag{9}$$

With the increase of j, the proportion of NRA gradually increases, and the proportion of RC gradually decreases.

Fig. 1 illustrates the concepts of decomposition, neighborhoods and groups in our proposed MOACO/D-Net algorithm. Ant i is responsible for solving the i^{th} subproblem, and is associated with a weight vector $\boldsymbol{\lambda}^i$ and an objective function $g(\boldsymbol{x}|\boldsymbol{\lambda}^i)$. Since $\boldsymbol{\lambda}$ in $g(\boldsymbol{x}|\boldsymbol{\lambda})$ is a continuous variable, two neighboring subproblems with close weight vectors are likely to have similar solutions. Based on this observation, the concepts of Neighborhood and Group in MOACO/D-Net are defined as follows.

- **Neighborhood:** Each of the N ants for the N subproblems in MOACO/D-Net is associated with a corresponding weight vector $\boldsymbol{\lambda}^i$. The neighborhood $\mathcal{N}(i)$ of ant i consists of T ants, whose subproblems' weight vectors are closer to $\boldsymbol{\lambda}^i$ than the other weight vectors. We assume that $i \in \mathcal{N}(i)$, i.e. ant i is its own neighbor.

- **Group:** MOACO/D-Net aims at approximating the whole PF, thus a single pheromone matrix cannot guide all the ants towards the optimal solutions with different weight vectors simultaneously. The ant colony is thus divided into K groups, $K < N$. All the ants in the same group share the same pheromone matrix, and each group searches for optimal solutions which approximate a part of the whole PF.

- **External Archive:** An archive is defined to store the nondominated solutions found during the evolution. All the solutions dominated by a new solution *x* in the archive are removed after *x* is added.

3 The proposed MOACO/D-Net for community detection

To design an effective multi-objective ACO algorithm for the community detection problem, important issues such as how to represent the solution, and how to define the pheromone information and heuristic information need to be addressed. Other research issues include how to initialize and update the pheromone matrices, and how to define the maximum and minimum pheromone trails. In this section, the MOACO/D-Net algorithm will be introduced in details.

3.1 Representation

For community detection problem, one intuitive choice of representation is to assign different community identifier randomly to each gene of the solution with length of n, where n is number of vertices. Thus the vertices (genes) assigned with same community identifier belong to the same community. However, these vertices in the network may not have links with each other, thus resulting invalid solutions. Actually in practice an initialization operator is usually applied (Gong et al. 2011; Mu et al. 2015). In every chromosome with *n* dimensions, each vertex is put into a different community, thus the number of communities for each chromosome in the initial population is n. Then for each chromosome, a vertex is randomly selected and its community identifier is assigned to its adjacent vertices which have links with it. This operation is repeated several times for each chromosome to initialize the population. Another frequently used method is the locus-based adjacency representation (Handl and Knowles 2007). In this graph-based representation, each solution consists of *n* genes and each gene can take the index of one of its adjacent nodes as its allele value. Thus, a value of *j* assigned to the i^{th} gene is then interpreted as a link between node *i* and *j* in the resulting partition solution, and they will be in the same community. The decoding of this representation requires the identification of all connected components. All vertices belong to the same connected component are then assigned to one community. Note that, this decoding step can be done in linear time by using a simple backtracking scheme (Shi et al. 2009).

In our proposed algorithm, new solutions are constructed by ants. Due to the property of ACO algorithm, pheromone trails should be laid on the edges of the best solution, and the encoding scheme should keep the information of locus. Therefore the locus-based adjacency representation is adapted to encode the solution. Furthermore the process of locus-based encoding scheme does not need the knowledge of the number of communities in advance.

Using the locus-based representation, for a particular network $G = (V, E)$ with $n = |V|$ vertices, a solution $\boldsymbol{x}$ is encoded as a vector $(x_1, x_2, \dots, x_n)$ with *n* genes. If the allele value of the i^{th} element x_i is *j*, then vertices v_i and v_j are connected and they both belong to the same community.

Fig. 2 shows an illustrative example of the locus-based adjacency scheme. Fig. 2(a) is an example network with 11 vertices and 16 edges. Fig. 2(b) provides a possible encoded solution (3 1 2 6 4 7 8 5 10 11 9), where the allele value of the 1^{st} element is 3, meaning v_1 and v_3 are in the same community, and the 2^{nd} element is 1, meaning v_2 and v_1 are in the same community, etc. A directed graph is then constructed as Fig. 2(c) to demonstrate the procedure of decoding, directing v_1 to v_3, and v_2 to v_1, etc. The network is thus partitioned into three clusters: A, B and C. A contains 3 vertices $\{1,2,3\}$, B contains 5 vertices $\{4,5,6,7,8\}$, and the remaining vertices $\{9,10,11\}$ belong to C.

3.2 Initialization

The ant colony is initialized with N ants, and each ant i represents a solution $\boldsymbol{x}^i$ corresponding to subproblem i. The weight vector of $\boldsymbol{x}^i$, i.e., $\boldsymbol{\lambda}^i$ is calculated according to Eq. (7). $\{\boldsymbol{F}^1, \boldsymbol{F}^2, \dots, \boldsymbol{F}^N\}$ is the set of objective function vectors of $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^N\}$, where $\boldsymbol{F}^i = \boldsymbol{F}(\boldsymbol{x}^i)$ as defined in Eq. (6).

The neighborhood $\mathcal{N}(i)$ of ant i is obtained as follows. The Euclidean distances d_{ij} between $\boldsymbol{\lambda}^i$ and $\boldsymbol{\lambda}^j$, $j \in \{1,2,\dots,N\}$, are calculated and sorted, and then T ants with T smallest distance are selected to form $\mathcal{N}(i)$, i.e.:

$$\mathcal{N}(i) = \{\boldsymbol{x}^j \mid d_{ij} \leq d_T, j = 1,2,\dots,N\} \tag{10}$$

where d_T is the T^{th} smallest value among all the d_{ij}, $j = 1,2,\dots,N$.

K pheromone matrices $\{\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \dots, \boldsymbol{\tau}^K\}$ are set for the K groups. The pheromone trails on edge (k,l) in group j are initialized with $\tau_{kl}^j = \tau_0$, where τ_0 is a sufficiently large value.

The heuristic matrix $\boldsymbol{\eta}$ is defined by Pearson correlation. $\eta_{kl} = 0$ if there is no connection between vertex k and vertex l, otherwise η_{kl} is defined and initialized as follows by Eq. (11).

$$\eta_{kl} = \frac{1}{1 + e^{-C(k,l)}} \tag{11}$$

$$C(k,l) = \frac{\sum_{V_s \in V}(A_{ks} - \mu_k)(A_{ls} - \mu_l)}{n\sigma_k\sigma_l} \tag{12}$$

In (11), $C(k,l)$ is the Pearson correlation between vertex k and vertex l in G, which indicates the structure similarity of vertex k and vertex l; in (12), $\boldsymbol{A}$ is the adjacency matrix of G; $\mu_k = \sum_s A_{ks}/n$ and $\sigma_k = \sqrt{\sum_s (A_{ks} - \mu_k)^2/n}$. Constructed based on Pearson correlation, η contains heuristic information. The larger the value of η_{kl} is, the more similar in structure vertex k and vertex l are and they are more likely to belong to the same community. η_{kl} always takes values in range (0, 1).

For each ant i, the corresponding solution is initialized by the locus-based adjacency scheme and each allele value l assigned to the k^{th} gene is randomly determined from the adjacent vertices of vertex k. All the nondominated solutions are thus extracted from the N solutions and stored in the external archive.

3.3 Solution construction

In traditional ACO, the new solution is constructed by the transition probability model. The transition probability model usually contains the pheromone matrix and the heuristic matrix. The transition probability model should be built according to the problem to be solved. For example, to solve traveling salesman problem, the transition probability generally should be updated according to the length of the shortest path found so far, and the heuristic matrix should consider the distances between the current city and its neighboring cities. To solve the community detection problem using a multi-objective ACO algorithm based on decomposition, a new transition probability model has to be built by considering the characteristics of community detection problem.

Assume that ant $\boldsymbol{x}^i = (x_1^i, x_2^i, \dots, x_n^i)$ is in group j, where $j = ceil(i \times K/N)$. For $i = 1,2,\dots,N$, ant i constructs a new solution $\boldsymbol{y}^i = (y_1^i, y_2^i, \dots, y_n^i)$ using a pseudo-random transition probability model. This model involves three factors: the pheromone information τ_{kl}, the heuristic information η_{kl} and the current solution $\boldsymbol{x}^i$. The transition probability model is presented in Eqs. (13)-(16).

$$P_{kh}^i = \begin{cases} \dfrac{\phi_{kh}}{\sum_{s \in \mathcal{N}v(k)} \phi_{ks}}, & h \in \mathcal{N}v(k) \\ 0, & otherwise \end{cases} \tag{13}$$

$$\phi_{kh} = \left[\tau_{kh}^j + \Delta \times In(\boldsymbol{x}^i, (k,h))\right]^\alpha \cdot (\eta_{kh})^\beta, \qquad h \in \mathcal{N}v(k) \tag{14}$$

$$\Delta = \frac{1}{1 + g(x^i|\boldsymbol{\lambda}^i)} \tag{15}$$

P_{kh}^i in Eq. (13) is the probability for vertex k to choose vertex h in subproblem i. $\mathcal{N}v(k)$ is the set of adjacent vertices of vertex k. α and β are two control parameters which aim at striking a balance between pheromone value and heuristic information. τ_{kh}^j is the element in the pheromone matrix of group j. In Eq. (14), $In(\boldsymbol{x}^i, (k,h))$ equals to 1 if edge (k,h) is in x^i, otherwise it equals to 0. $g(\boldsymbol{x}^i|\boldsymbol{\lambda}^i)$ in Eq. (15) denotes the *g-value* of $\boldsymbol{x}^i$ as defined in Eq. (7) with the weight vector $\boldsymbol{\lambda}^i$. Then the better $\boldsymbol{x}^i$ is in term of *g-value*, the more likely the vertex *h* is selected as the allele value of the *k*th element of the new solution as edge (k,h) is in x^i.

A new solution $\boldsymbol{y}^i = (y_1^i, y_2^i, \dots, y_n^i)$ is constructed by generating each allele value of each element according to the transition probability model above. The allele value l of the *k*th element in the new solution is constructed as Eq. (16).

$$l = \begin{cases} \underset{h \in \mathcal{N}v(k)}{\operatorname{argmax}}(\phi_{kh}), & rand < r \\ l', & otherwise \end{cases} \tag{16}$$

where rand is a uniformly distributed random number within (0, 1). If rand is smaller than *r*, which is a preset positive threshold smaller than 1, vertex *l* with the largest ϕ value in Eq. (14) is selected as the allele value of the *k*th element; otherwise vertex l is determined according to the probability P_{kh} in Eq. (13) using the roulette wheel selection. *r* in Eq. (16) is applied to simplify the transition probability model. If *r* = *0*, Eq. (16) will degrade into a traditional transition probability model

3.4 Solution improvement

To achieve better performance, it is necessary to improve the solution constructed. We design three different improvement operator named Improvement-I, Improvement-II and Improvement-III. The proposed algorithm adopting Improvement-I, Improvement-II or Improvement-III are called MOACO/D-Net-I, MOACO/D-Net-II, and MOACO/D-Net-III respectively. We will compare MOACO/D-Net-I, MOACO/D-Net-II and MOACO/D-Net-III in the experiments, and adopt the best one (MOACO/D-Net-III) as the final version of MOACO/D-Net.

3.4.1 Improvement-I

Input the solution $\boldsymbol{s}$ constructed by ant *i* to Improvement-I. Repeat the procedure of solution construction for $(N_{\text{ant}} - 1)$ times. Then ant i chooses the best solution from N_{ant} new solutions to be the output $\boldsymbol{x}^i$ according to the $g(\boldsymbol{x}^i|\boldsymbol{\lambda}^i)$ values of the N_{ant} solutions.

3.4.2 Improvement-II

In the network, the nodes which belong to one community but have connections with other nodes belonging to the adjacent communities can be called edge nodes. These nodes are most possibly become misclassified. We has proposed an error recovery operator to improve the quality of a given solution for community detection by checking and repair the edge nodes (Mu et al. 2014), which was proved to be effective. Inspired by this idea, we design two improvement operators to check and repair the edge nodes of the solutions that are constructed by the ants and thus improve their quality. The main advantages of these improvement operators over Improvement-I lie in that they are simple and cost less time. More importantly, these improvement operators do not cost any fitness evaluations during the process of execution.

Improvement-II is designed as follows. Based on the definition of weak community, define a measure of the two adjacent communities' denseness as Eq. (17):

$$Mc(c_1, c_2) = \frac{\sum k_{in} - \sum k_{out}}{\sum k} \tag{17}$$

where c_1, c_2 are two adjacent communities, and $\sum k$ is the sum of the degrees of all nodes in c_1 and c_2; $\sum k_{in}$ is the sum of the internal degree of all nodes in c_1 and c_2; $\sum k_{out}$ is the sum of the external degree of all nodes in c_1 and c_2. Assume a node v_j originally belongs to c_1. If the value of $Mc(c_1, c_2)$ increases after v_j moves from c_1 to c_2, it indicates v_j is more likely to belong to c_2, and assigning v_j to c_2 may result in a better solution..

Input the solution $\boldsymbol{s}$ constructed by ant *i* to Improvement-II, and the operator first find out all the edge nodes in $\boldsymbol{s}$. For the edge node v_j $(j = 1, 2, \dots, l$ and *l* is the total number of edge nodes), find out all the adjacent communities c_p $(p = 1, 2, \dots, num\_c$, and num_c is the total number of the adjacent communities) which have links with v_j. Initially, the community of v_j in $\boldsymbol{s}$ is c_j. For one adjacent community c_p that have links with v_j, calculate $Mc_{pre}(c_j, c_p)$ using Eq. (17). Then assume v_j is in c_p and calculate $Mc_{cru}(c_j, c_p)$ using Eq. (17). If $Mc_{cru}(c_j, c_p) > Mc_{pre}(c_j, c_p)$, assign v_j from c_j to c_p and c_p becomes the new c_j. After repeating for num_c times, v_j is assigned to a more appropriate community. After repeating above procedure for *l* times, Improvement-II is terminated and the resulting solution $\boldsymbol{s}'$ is output as the new solution $\boldsymbol{x}^i$ of ant *i*.

3.4.3 Improvement-III

Improvement-III is designed as follows. Based on the definition of strong community, define a measure of the node's denseness as Eq. (18):

$$Md^j = \frac{k_{in}^j - k_{out}^j}{k^j} \tag{18}$$

where k^j is the degree of node *j*; k_{in}^j is the internal degree of node *j*; k_{out}^j is the external degree of node *j*. Assume one edge node v_j has two adjacent communities, C_A and C_B. Assume v_j is in C_A, and calculate its Md_A^j for C_A; Then assume v_j is in C_B, and calculate its Md_B^j for C_B; If $Md_A^j > Md_B^j$, it indicates that v_j is more likely belong to C_A based on the definition of strong community.

Input the solution $\boldsymbol{s}$ constructed by ant *i* to Improvement-III, and the operator first find out all the edge nodes in $\boldsymbol{s}$. For the edge node v_j $(j = 1, 2, \dots, l$, and *l* is the total number of edge nodes), find out all the adjacent communities that have links with v_j. Calculate Md^j of v_j for all the adjacent communities one by one, assuming v_j is in the current adjacent community every time. Then assign v_j to the community with the largest Md^j. After repeating above procedure for *l* times, Improvement-III is terminated and the resulting solution $\boldsymbol{s}'$ is output as the new solution $\boldsymbol{x}^i$ of ant *i*.

3.5 Update of the external archive

After all the N new solutions $\boldsymbol{y}^1, \boldsymbol{y}^2, \dots, \boldsymbol{y}^N$ are constructed, $\boldsymbol{F}(\boldsymbol{y}^i)$ is calculated, where $i = 1,2,\dots,N$, and is compared with the objective vectors of nondominated solutions in the external archive. If no solution in the archive dominates $\boldsymbol{y}^i$, $\boldsymbol{y}^i$ is added to the archive. Those solutions dominated by $\boldsymbol{y}^i$ are removed.

3.6 Update of pheromone matrices

The pheromone matrix is updated according to those ants with a good performance or better objective values. In the single-objective ACO algorithms, the iteration-best or global-best solution determined by the single objective is usually used to update a single pheromone matrix. In this paper, for $j = 1,2,\dots,K$, one pheromone matrix $\boldsymbol{\tau}^j$ of group j is updated by the information extracted from those good new solutions in group j using Eq. (19).

$$\tau_{kl}^{j}(t+1) = \rho \cdot \tau_{kl}^{j}(t) + \sum_{\boldsymbol{x}^{i} \in \mathcal{H}(j)} \frac{1}{1+g(\boldsymbol{x}^{i}|\boldsymbol{\lambda}^{i})} \times In\big(\boldsymbol{x}^{i},(k,l)\big) \tag{19}$$

where t is the iteration index, ρ is the persistence rate and $\mathcal{H}(j)$ is the set of good new solutions in group j which are nondominated during the update of the external archive and have been chosen to be added into the external archive. The second item in Eq. (19) means pheromone trails are only laid on the edge (k,l) in x^i.

Following the Max-min ant system (Stützle and Hoos 2000), the pheromone of each element in the pheromone matrices is limited to a range of $[\tau_{min}, \tau_{max}]$, where τ_{max} is determined by an estimate of asymptotically maximum value shown in Eq. (20).

$$\tau_{max} = \frac{B+1}{(1-\rho)(1+g_{min})} \tag{20}$$

where B is the number of nondominated solutions found at the current iteration, i.e. the number of solutions added to the archive at the current iteration. g_{min} is the smallest value of g for all the N subproblems. τ_{max} is used to avoid that some edges of pheromone matrices accumulate too much pheromone and thus leading to premature.

τ_{min} is set using Eq. (21).

$$\tau_{min} = \varepsilon \cdot \tau_{max} \tag{21}$$

where ε is a fairly small number (ε is set equal to 0.005 in this paper).

After each iteration, τ_{min} and τ_{max} are updated according to Eqs. (20) and (21). Therefore the procedure of updating pheromone trails contains two stages: a weakening stage and a reinforcement stage. The first stage is for global updating, where the values of pheromone on all the edges are reduced by the evaporation factor, aiming to evaporate pheromone and avoid a premature convergence. During the second stage, all the edges in the new nondominated solutions are reinforced. This strategy makes sure that the information of good solutions is retained to promote these ants to the next iteration for finding better solutions.

3.7 Update of $\boldsymbol{x}^i$

Before the next generation, for $i = 1,2,\dots,N$, each individual ant i then updates the current solution x^i according to its neighbors. Based on its own objectives $\boldsymbol{F}(\boldsymbol{x}^i)$, if there is a solution $\boldsymbol{x}^*$ with the smallest g-value in the neighborhood of ant i which satisfies $g(\boldsymbol{x}^*|\boldsymbol{\lambda}^i) < g(\boldsymbol{x}^i|\boldsymbol{\lambda}^i)$, and $\boldsymbol{x}^*$ has not been used to replace any other old solutions, then $\boldsymbol{x}^i$ is replaced by $\boldsymbol{x}^*$. This strategy encourage ants to exchange information with its neighbors, even the two neighbors are not in the same group. The update of $\boldsymbol{x}^i$ thus strengthens the cooperation between different ant groups.

The complete pseudo-code of the proposed algorithm MOACO/D-Net for community detection is depicted in Algorithm 1.

Algorithm 1: MOACO/D-Net

```
Input: A complex network modeled as $G = (V, E)$
Step 1: Initialization
    Set parameters $N, \tau_0, \alpha, \beta, \rho, r, K, T, \varepsilon, Gen$;
    Initialize $K$ pheromone matrices $\{\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \dots, \boldsymbol{\tau}^K\}$, heuristic information matrix $\boldsymbol{\eta}$
    Initialize $N$ ants $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^N\}$, calculate objectives $\{\boldsymbol{F}(\boldsymbol{x}^1), \boldsymbol{F}(\boldsymbol{x}^2), \dots, \boldsymbol{F}(\boldsymbol{x}^N)\}$;
For $i$ = 1 to $Gen$
Step 2: Solution Construction
```

```
    For $i$ = 1 to $N$
        Construct a new solution $\boldsymbol{s}$ according to Eqs. (13) to (16);
        $\boldsymbol{x}^i \leftarrow$ Improvement operator ($\boldsymbol{s}$)
            // Call the subprogram Improvement-I, Improvement-II or Improvement-III
    End For
Step 3: Update the External Archive
     For $i$ = 1 to $N$
         If (no solution in the archive dominates $\boldsymbol{x}^i$)
             Add $\boldsymbol{x}^i$ to archive;
             Remove all the solutions dominated by $\boldsymbol{x}^i$;
     End For
Step 4: Update the Pheromone Matrices
      Update $\{\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \ldots, \boldsymbol{\tau}^K\}$, $\tau_{max}$, and $\tau_{min}$ according to Eqs. (19) to (21);
Step 5: Update $\boldsymbol{x}^i$
      For $i$ = 1 to $N$
          If $(g(\boldsymbol{x}^*|\boldsymbol{\lambda}^i) < g(\boldsymbol{x}^i|\boldsymbol{\lambda}^i))$ & $\boldsymbol{x}^*$ has not been used //$\boldsymbol{x}^* \in \mathcal{N}(i)$
          $\boldsymbol{x}^i \leftarrow \boldsymbol{x}^*$;
      End For
End For
Output: External archive, Pareto front
        // Pareto front is the set of objective function vectors in Eq. (6) of solutions in external archive
// Subprogram Improvement-I
Input: $\boldsymbol{s}$
    $S = \emptyset$; $S \leftarrow S \cup \boldsymbol{s}$
    For $j$ = 1 to $N_{ant} - 1$ // $N_{ant} - 1$ solutions are generated by ant i
        Construct a new solution $\boldsymbol{s}$ according to Eqs. (13) to (16);
        $S \leftarrow S \cup \boldsymbol{s}$;
    End For
        $\boldsymbol{x}^i \leftarrow Best(S)$;
Output: $\boldsymbol{x}^i$
// Subprogram Improvement-II
Input: $\boldsymbol{s}$
      $\boldsymbol{x}^i \leftarrow \boldsymbol{s}$; $\boldsymbol{v} = (v_1, v_2, \ldots, v_l) \leftarrow$ Find out the edge nodes in $\boldsymbol{x}^i$ // $l$ is the number of edge nodes
      For $j$ = 1 to $l$ // Repair $v_j$ in $\boldsymbol{x}^i$
          $c_j \leftarrow$ the community containing $v_j$ in $\boldsymbol{x}^i$
          For p = 1 to $num\_c$ // $num\_c$ is the number of the adjacent communities linked with $v_j$
              $c_p \leftarrow$ the current adjacent community of $v_j$
              $Mc_{pre}^j \leftarrow$ Calculate $Mc^j(c_j, c_p)$ according to Eq. (17)
              assume $v_j$ is in $c_p$
              $Mc_{cur}^j \leftarrow$ Calculate $Mc^j(c_j, c_p)$ according to Eq. (17)
              If $Mc_{cur}^j > Mc_{pre}^j$
                  assign $v_j$ to $c_p$ in $\boldsymbol{x}^i$
                  $c_j \leftarrow c_p$
              End If
          End For
        End For
Output: $\boldsymbol{x}^i$
// Subprogram Improvement-III
Input: $\boldsymbol{s}$
        $\boldsymbol{x}^i \leftarrow \boldsymbol{s}$
        $\boldsymbol{v} = (v_1, v_2, \ldots, v_l)$ ←Find out the edge nodes in $\boldsymbol{x}^i$ // $l$ is the number of edge nodes
        For $j$ = 1 to $l$ // Repair $v_j$ in $\boldsymbol{x}^i$
            max_$Md^j \leftarrow 0$
            For $p$ = 1 to $num\_c$ // $num\_c$ is the number of the adjacent communities linked with $v_j$
                Calculate $Md_p^j$ after assuming $v_j$ is in the current community according to Eq. (18) ;
                If $Md_p^j >$ max_$Md^j$
                    max _$Md^j \leftarrow Md_p^j$
                End If
            End For
            assign $v_j$ to the community with max_$Md^j$ in $\boldsymbol{x}^i$
        End For
Output: $\boldsymbol{x}^i$
```

4 Experimental results

In this section analysis on the results of experiments on LFR benchmark networks and eight real-world networks are presented in detail. The proposed algorithm is simulated using MATLAB 2018a. All the experiments have been performed on the same computer with an Inter(R) Xeon(R) E5-2630 v4 CPU, 2.20 GHz, 64GB RAM under Windows 10 OS.

The data sets used in our experiments include LFR benchmark networks and eight real-world networks as follows.

- **LFR benchmark network**: the Lancichinetti-Fortunato-Radicchi (LFR) benchmarks (Lancichinetti et al. 2008) are widely used for generation of synthetic networks. In LFR benchmarks, a rich set of parameters can be used to generate different types of network topology. The parameter setting for the LFR benchmarks is listed in Table 1.
- **Zachary's karate club network**: consists of 34 vertices and 78 edges, introduced by Zachary (1977). Zachary observed a karate club with 34 members over a period of two years, and found that there is a disagreement between the administrator and the instructor of the club. The club is separated into two groups ultimately. The network splits naturally into two communities.
- **Dolphin social network**: was constructed by Lusseau et al. (2003), who observed 62 bottlenose dolphins' behavior during seven years in Doubtful Sound, New Zealand. A link between two dolphins means they have a more frequent association statistically. The dolphin network includes 159 edges and is divided into two groups according to their gender.
- **American college football network**: includes 115 vertices and 613 edges. The football network was established by Girvan and Newman (2002). It represents American football games between Division I colleges during the 2000 season. Vertices in the network represent teams and edges represent games between two teams. The network can be partitioned into 12 clusters.
- **Books on US politics (Polbooks network)**: is a network of books on politics discovered by Krebs and divided by Newman (2006). It consists of 105 vertices and 441 edges. Each vertex in this network represents a book from Amazon.com on American politics, and each edge connects two books which are frequently co-purchased by the same buyer. The books are classified into three classes according to the descriptions and reviews of the books.
- **SFI network**: represents 271 scientists in residence at the Santa Fe Institute during any part of calendar year1999

or 2000, and their collaborators. An edge is drawn between a pair of scientists if they coauthored one or more articles during the same time period. The biggest part of the SFI networks consist of 118 nodes and 200 edges. We only do experiments on this part.

- **Netscience network**: represents the co-authorship of scientists working on network theory and experiment. This network consists of 1461 nodes and 2742 edges. It is weighted and we handle it as an unweighted one in our experiments.
- **Protein network**: is an undirected network containing protein interactions in yeast. A node represents a protein and an edge represents a metabolic interaction between two proteins. This network consists of 1870 nodes and 2277 edges.
- **US Power Grid network**: is an undirected network containing information about the power grid of the Western States of the United States of America. An edge represents a power supply line. A node is either a generator, a transformator or a substation. This network consists of 4941 nodes and 6594 edges.

Table 1 Parameter setting of LFR benchmarks.

Parameters	Non-overlapping
Network size N	{100,200,300,400,500,600,700,800,900,1000}
Mixing parameter μ	{0,0.05,0.1,0.15,0.25,0.3,0.35,0.4,0.45,0.5}
Average degree $\overline{k}$	20
Maximum degree k_{max}	50
Exponent of nodes' degree distribution τ_1	2
Exponent of community size distribution τ_2	3

Table 2 Real-world networks in the test.

Network	*n* (number of vertices)	n_e (number of edges)	True partitions
karate network	34	78	Known with 2 clusters
Dolphin network	62	160	Known with 2 clusters
Football network	115	613	Known with 12 clusters
Polbooks network	105	441	Known with 3 clusters
SFI	118	200	Unknown
Netscience	1461	2742	Unknown
Protein	1870	2277	Unknown
US Power Grid	4941	6594	Unknown

4.1 Evaluation criteria

Two evaluation criteria are introduced to measure the quality of the partition. One of the most commonly used evaluation criteria is modularity Q as shown in Eq. (22).

$$\mathrm{Q} = \frac{1}{2n_e}\sum_{ij}\left(\left(A_{ij} - \frac{k_i k_j}{2n_e}\right) \times \delta\big(In(i), In(j)\big)\right) \tag{22}$$

where $n_e = \frac{1}{2}\sum_{ij} A_{ij}$ is the number of edges in the network, A is the adjacency matrix, $k_i = \sum_j A_{ij}$ is the degree of vertex i, $In(i)$ is the index of the community that vertex i belongs to, and the Kronecker delta function $\delta(u, v) = 1$ if $u = v$, otherwise $\delta(u, v) = 0$.

The modularity of a network should be maximized during the evolution. Actually the Q values of most social networks fall in the range of 0.3 to 0.7 (Newman 2004). Generally the partition corresponding to its maximum on a given network should be the best, or at least is a very good one. This principle is the main motivation for modularity optimization in the most popular algorithms to find a more accurate structure of the networks.

Another important criterion is the normalized mutual information (NMI) (Danon et al. 2005), which is defined to measure the similarity between community structure found by the proposed algorithm and the true partition. Assume there are two different partitions for a given network: partition A and partition B. A confusion matrix Z is defined, whose element Z_{ij} is the number of same vertices in community i of partition A and community j of partition B. The NMI can then be evaluated by Eq. (23):

$$\mathrm{NMI(A,B)}=\frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}Z_{ij}\log\left({Z_{ij}n}/{Z_{i.}Z_{.j}}\right)}{\sum_{i=1}^{C_A}\left(Z_{i.}\log\left({Z_{i.}}/{n}\right)\right)+\sum_{j=1}^{C_B}\left(Z_{.j}\log\left({Z_{.j}}/{n}\right)\right)} \tag{23}$$

where C_A is the total number of communities in partition A. $Z_{i.}=\sum_j Z_{ij}$, which means the sum of elements in row i, $Z_{.j}=\sum_i Z_{ij}$, and n is the total number of vertices.

The partition which is more closer to the real one has a larger NMI value. Obviously $\mathrm{NMI}(A,B)=\mathrm{NMI}(B,A)$ and $\mathrm{NMI}(A,B)=1$ if $A=B$. Generally NMI is also limited in the range of $[0,1]$.

Therefore the values of Q and NMI are proper measures of a solution. However there does not exit a positive correlation between Q and NMI. In some cases, a good solution with a larger Q may not share a larger NMI, and a solution with a larger NMI may not share a larger Q either.

4.2 Comparison of three variants of MOACO/D-Net and the analysis of run-time

In this section, we will compare and analyze three variants of the proposed algorithm MOACO/D-Net, i.e., MOACO/D-Net-I, MOACO/D-Net-II, and MOACO/D-Net-III, including their performance and run-time. To be fair, the same set of parameters are used for three variants according to experience: N=60, α=1, r=0.1, K=5, T=8, β=4, ρ=0.95; N_{ant}=50.

4.2.1 Comparison on four real-world networks with true partition known

This section provides the comparison on four real-world networks whose true partitions are known, i.e., the Zachary's karate club, the Dolphin social network, the American College football, and the Books on US politics. The run-time of three variants is also compared and analyzed. For each network, we calculate average values of NMI and Q obtained by each algorithm over 20 independent runs.

The experimental results are shown in Table 3. It can be observed from Table 3 that the Q values and the NMI values obtained by MOACO/D-Net-III are a little better than those by the others on three out of four networks, meanwhile the run-time of MOACO/D-Net-III is significantly shorter. Among three variants, MOACO/D-Net-I adopting Improvement-I is the most time consuming. For ant i, Improvement-I needs to produce N_{ant} solutions, then calculate $g(\boldsymbol{x}^i|\boldsymbol{\lambda}^i)$ values of them and select the best one as the final new solution of ant i. This procedure is very time consuming. In contrast, each time Improvement-II and Improvement-III only repair one solution for ant i, which are simple and cost less time.

Table 3 Results of three variants of MOACO/D-Net on four real-world networks

Metric	Networks	**MOACO/D-Net-I**	**MOACO/D-Net-II**	**MOACO/D-Net-III**
$\mathrm{Q_{avg}}$	**Karate**	0.420	0.420	0.419
	Dolphins	0.523	0.523	**0.525**
	Football	0.603	0.603	**0.604**
	Polbooks	0.517	0.525	**0.526**
$\mathrm{NMI_{avg}}$	**Karate**	1	1	**1**
	Dolphins	0.981	0.988	**1**

	Football	0.927	0.923	**0.927**
	Polbooks	0.583	0.567	**0.606**
Time(s)	**Karate**	58.797	10.224	**8.232**
	Dolphins	116.842	23.977	**16.761**
	Football	340.412	77.999	**45.674**
	Polbooks	248.112	52.226	**37.759**

4.2.2 Relation between run-time and network size

To analyze the relation between run-time of three improvement operators and the size of network, the LFR benchmark networks are tested here, using the parameters shown in Table 1 and the network size changes from 100 to 1000. We record the time of generating N (N=60) new solutions by three improvement operators in one generation on different networks over 10 runs. The average results are shown in Fig. 3. It can be seen from Fig. 3 that the time of Improvement-III increases most slowly with the increasing of network size. To deal with large scale network, MOACO/D-Net-III adopting Improvement-III is a reasonable choice. Then we test MOACO/D-Net-III on the same LFR benchmark networks with size changing from 100 to 1000 over 10 independent runs, and the average run-time is shown in Fig. 4. So MOACO/D-Net-III is able to deal with large scale network within acceptable time.

4.2.3 Effectiveness of improvement operator: Improvement-III

In this section, we will show whether Improvement-III really benefits the performance of MOACO/D-Net. Therefore we delete the operator of Improvement-III from MOACO/D-Net-III, and call this temporary version as MOACO/D-Net-X, i.e., there is no improvement operator used in MOACO/D-Net-X. Then MOACO/D-Net-X and MOACO/D-Net-III are tested on the LFR benchmark networks with same size of 1000 and different mixing parameters over 10 independent runs. It can be found from Fig. 5 that NMI values obtained by two versions decease with the increasing of mixing parameter. Because when the mixing parameter increases, it becomes more difficult to find the true partition of networks. However, NMI values obtained by MOACO/D-Net-III are always larger than those by MOACO/D-Net-X, and especially for the situation with large mixing parameter, the advantages of MOACO/D-Net-III is more distinct. These indicate that Improvement-III does benefit the performance of MOACO/D-Net.

Therefore in the subsequent experiments, MOACO/D-Net-III is used as the final version of the proposed algorithm, written as MOACO/D-Net.

4.3 Analysis and experiments on parameters

It was observed that parameters of ACO algorithm in solving one problem often are different from those for solving another problem. The ranges of the optimal parameters are mostly determined based on experience. For the community detection problem, priori knowledge about parameters in ACO is rare, and an analysis on the four essential parameters in the ACO algorithm is presented in the following paragraphs. The main parameters are as follows.

- α(pheromone information factor) and β(heuristic information factor): The value of α reflects the relative importance of the pheromone trail. When α is large, the current ant is more likely to choose the route taken by the ants in the previous generation. β reflects the relative importance of heuristic information and adjusts the intensity of utilizing the priori information during the search. A larger β means that the ant is more likely to select local optimal components. This will promote convergence speed, however, may cause premature (i.e. the search is trapped in local optimum at early stage of evolution). The values of α and β determine whether the ant pays more attention to pheromone information provided by good solutions or heuristic information provided by the structure similarity of vertices in the search process. The value of α is always set equal to 1 in our experiments, and the proportion of α and β is controlled by the value of β. The default value of β is set equal to 6 in our experiments.
- ρ (persistence rate of pheromone trail): ρ is defined as persistence rate of pheromone trail. (1-ρ) is thus the evaporation rate, which models the phenomenon that the pheromone trail on the route to the goal evaporates as time goes on. ρ is needed to

avoid a too rapid convergence of the algorithm towards a local optimal region. It implements a useful forgetting mechanism, preferring exploration of new areas in the search space. The default value of ρ is 0.95 in our experiments.

- r (threshold value in solution construction): This is a positive value in the pseudo-random transition probability model. A proper value of r can speed up the search procedure and simplify the algorithm by adopting a greedy strategy to construct the solution. When r is too large, each ant is more likely to choose an edge with a larger value of pheromone trail and structure similarity, thus search may be trapped to local optimal. On the other hand, when r is very small, especially when $r = 0$, the pseudo-random transition probability model degrades into a traditional transition probability model. The default value of r is 0.2 in our experiments.

In the MOACO-Net algorithm, there exist another two tunable parameters that also may affect the Pareto optimal solutions.

- K(number of groups) and T(number of neighbors of each ant): These two special parameters exist in the multi-objective ACO algorithm only. There is no theoretical analysis on the performance of ant colony with different number of groups and different number of neighbors so far. For a certain ant colony, K is the number of pheromone matrices to be established, and T influences the searching scope during the update of current solution x^i.

The purpose of our preliminary experiments on these parameters is to find a proper range of values for every parameter, respectively. All the experiments about different parameters have been done on four real-world networks over 10 independent runs, setting ant colony size equal to 60 and the maximum number of iterations equal to 50.

To analyzeβ, we empirically set $\alpha = 1$, $\rho = \{0.7,0.75,0.8,0.85,0.9,0.95\}$, and $\beta = \{0.2,0.5,0.7,1,2,4,6,8,10,12,14\}$. The results are shown in Fig. 6, and the results on Karate network are not given because it is easy to divide Karate network and its results cannot reflect the trend of changes. It can be observed from Fig. 6, that the values of NMI and Q are better when $\beta = \{2, 4, 6\}$. It indicates that medium values of β are reasonable. Too small β means useful local heuristic information is ignored, resulting degraded search speed and performance. While too large β means too much local heuristic information is used during the search, resulting that the algorithm is trapped into the local optimum.

The results of different values of ρ are presented in Fig. 7. We set $\alpha = 1$,$\rho = \{0.7,0.75,0.8,0.85,0.9,0.95\}$ and $\beta = \{0.2,0.5,0.7,1,2,4,6,8,10,12,14\}$. The bar charts show that Q and NMI for the Dolphins network, Football network and Politics books network are larger at $\rho = \{0.85,0.9, 0.95\}$ on the whole. Larger ρ means larger persistence rate of pheromone trail, and more helpful experience of good solutions is utilized during the search, thus leading to better performance.

From Fig. 8, we can find that NMI and Q are better when r is a small value, and both NMI and Q will reduce gradually when r increases. In Fig. 8 (a) for the Karate network, the watershed values for NMI and Q are $r = 0.2$ and $r = 0.9$, respectively. In Fig. 8 (b) for the Dolphins network, NMI and Q begin to fall at $r = 0.1$ and $r = 0.8$, respectively. In Fig. 8 (c) for the Football network, NMI and Q almost has the same value when $r \leq 0.9$. In the last network, NMI and Q start to reduce at $r = 0.1$ and $r = 0.5$ respectively.

From the results in Fig. 6 to Fig. 8, several conclusions about the parameters can be drawn. α and β adjust the ratio between the pheromone information and heuristic information together. A partition with a larger modularity Q, however, may not share a larger NMI. Therefore, the value of β still needs being chosen carefully. Larger persistence rate of pheromone trail ρ generally results in larger modularity Q, which means previous good experience is really helpful to the search. The threshold value r in solution construction allows the ant to choose an edge with a larger value of pheromone trail and structure similarity when it is larger than 0, which may speed up the search procedure and simplify the solution construction. However, larger r may cause the search to be trapped to local optimal. So small r is a better choice. Based on the observation from Fig. 6 to Fig. 8, the most appropriate ranges of α, β, ρ and r are introduced as follows: $\alpha = 1$, $\beta \in [\,2, 4, 6]$, $\rho \in [0.85,0.9, 0.95]$ and $r \in [0, 0.2]$ In the following experiments on K and T, we set $\{\alpha, \beta, \rho, r\}$ equal to the values: $\{1, 4, 0.95, 0.1\}$.

To analyze K, we examined $K = \{1, 3, 4, 5, 6, 10, 15, 20, 30, 60\}$. As shown in Fig. 9 there is no special fluctuation of both Q and NMI. The result in Fig. 10 for $T = \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$ shows quite a similar situation as that of Fig. 9. Therefore we can conclude that parameters K and T are not sensitive for these four real-world networks. As $K = 1$, all the ants are in one group and share the same pheromone matrix. It can be observed that results obtained by $K > 1$ are generally better that those obtained by $K = 1$. So dividing the ants into several groups is helpful for the algorithm. But if K is too large, the algorithm will take more time to record and update the pheromone matrices. T controls the number of neighbors of each ant and promotes information exchange between different ants. Since K and T are not sensitive, we prefer medium values of them.

Based on the analysis and experiments on parameters above, we obtain the parameter setting of MOACO/D-Net as shown in Table 4, and MOACO/D-Net will always adopt these parameters in the following experiments.

Table 4 Parameter setting of MOACO/D-Net

Parameter	Meaning	Value
N	Ant colony size	60
α	pheromone information factor	1
β	heuristic information factor	4
ρ	persistence rate of pheromone trail	0.95
r	threshold value in solution construction	0.1
K	number of groups	5
T	number of neighbors of each ant	8
ε	lower bound factor of pheromone information	0.005
Gen	the maximum number of iterations	50

4.4 Comparison with other algorithms on real-world networks with true partitions known

In this subsection the proposed algorithm is compared with MOEA/D-Net (Gong et al. 2012), MOGA-Net (Pizzuti 2009), IACO-Net (Mu et al. 2014) and FN (Newman 2004). In these experiments, the parameter setting of MOACO/D-Net is same with that shown in Table 4. Average results of the best Q and NMI over 20 independent runs obtained by MOEA/D-Net are shown in Table 6.

All the algorithms being compared in this subsection fall into three categories: FN is a greedy method maximizing the modularity; IACO-Net is ACO based algorithms with the single objective optimizing modularity; MOACO/D-Net, MOEA/D-Net and MOGA-Net are three different MOEAs. MOACO/D-Net and MOEA/D-Net optimize negative ratio association and ratio cut simultaneously, and MOGA-Net optimizes community score (*CS*) and community fitness (*CF*) simultaneously. For the sake of fairness, except FN, the results of the algorithms for comparison are from their original literatures which ensures that all the algorithms provide their results based on their best performance. Meanwhile, the main parameters of these algorithms except FN are listed in Table 5. Different from other population based algorithms, FN is a greedy method which runs in time $O\ ((n_e{+}n)n)$ and always gets the same result over different runs. It can be found that the *Number of evaluations* of MOACO/D-Net is much smaller than those of MOEA/D-Net and MOGA-Net, and is also smaller than that of IACO-Net. IACO-Net has a *self-learning* operator in each generation, so its real *Number of evaluations* is a little more than 5000.

Table 5 Parameters of different algorithms for comparison

Parameter	**MOACO/D-Net**	**MOEA/D-Net**	**MOGA-Net**	**IACO-Net**
Population size	60	100	300	50
the maximum number of iterations	50	400	30	100
Objective function	*NRA*	*NRA*	*CS*	Q

	and *RC*	and *RC*	and *CF*	
Number of evaluations	3000	40000	9000	>5000

It can be observed in Table 6 that IACO-Net obtained larger modularity values than FN over four networks, but interestingly, only over the football networks IACO-Net obtained a larger NMI value than FN. It is because that there is no strict positive co-relationship between the modularity value and the NMI value (Mu et al. 2015). In another word, the NMI value of a solution may be small when its modularity value is large. So maximizing modularity or any other single objective function, may not be the best choice for finding the good partitions of a network.

Table 6 Results of the proposed MOACO/D-Net and existing algorithms on four real-world networks

Metric	Networks	**MOACO/D-Net**	**MOEA/D-Net**	**MOGA-Net**	**IACO-Net**	**FN**
Q_{avg}	**Karate**	0.419	**0.420**	0.415	**0.420**	0.381
	Dolphins	0.525	0.520	0.505	**0.526**	0.495
	Football	**0.604**	**0.604**	0.515	0.602	0.550
	Polbooks	0.526	**0.527**	0.518	0.517	0.502
NMI_{avg}	**Karate**	**1**	**1**	**1**	0.687	0.837
	Dolphins	**1**	**1**	**1**	0.567	0.606
	Football	**0.927**	0.925	0.795	0.879	0.654
	Polbooks	**0.606**	0.596	0.597	0.523	0.534

For MOEAs, a group of nondominated solutions can be obtained in a single run, and the solution with the best modularity value and the solution with the best NMI value can be selected and recorded. Therefore three MOEAs performed better than the single-objective algorithms on the whole. As for three MOEAs, MOACO/D-Net and MOEA/D-Net obtained better values on modularity Q than MOGA-Net over four networks. For the Karate network and Dolphins network, all MOEAs obtained $NMI_{avg} = 1$, indicating that all MOEAs could always find the true partitions. For the Football network and Polbooks network, MOACO/D-Net obtained much better values on NMI than all the algorithms for comparison.

Then the statistic values of best modularity Q and the statistic values of best NMI obtained by MOACO/D-Net, IACO-Net and FN over the 20 runs on four real-world networks are shown in terms of box plots in Fig. 11 and Fig. 12. It can be found from Fig. 11 and Fig. 12 that the variability of Q values and NMI obtained by MOACO/D-Net over the 20 runs is relatively small compared to IACO-Net on each of the four networks except Dolphins network. NMI and Q values obtained by FN do not change over different runs because it is a greedy algorithm. As we can see from Fig. 12, NMI values obtained by MOACO/D-Net are much better than those by IACO-Net and FN. On the whole, the statistic results illustrate the effectiveness and stability of MOACO/D-Net.

4.5 Comparison with other algorithms on LFR Benchmark networks and four Real-world networks with true partitions unknown

In this section, we compare MOACO/D-Net with CNM (Newman 2004) and CNM-IC (Mu et al. 2014) on LFR benchmark networks with 1000 nodes and four real-world networks with true partitions unknown including SFI network, Netscience network, Protein network and US power grid network. The details of these networks are shown in Table 1 and Table 2. For each network, we computed the average NMI obtained by each algorithm over 20 independent runs. CNM performs a quick maximization of modularity Q and can handle large networks. CNM-IC is an improvement on CNM by adding a simple but effective operator.

The experimental results are shown in Fig. 13. It can be seen that NMI values obtained by MOACO/D-Net are always larger than those by CNM and CNM-IC as the mixing parameter increases. For modularity Q, MOACO/D-Net obtains larger values when μ is less than 0.35, and obtains smaller values than CNM and CNM-IC when μ is greater than 0.35. This situation is consistent with what have been observed before, i.e., larger Q value is not always leads to a better partition with larger NMI value. In LFR benchmark networks, each node shares a fraction $1 - \mu$ of its links with the other nodes within its community and a fraction μ with the other nodes outside its community in the network. So the higher the mixing parameter μ of a network is, the more difficult it is to reveal the community structure. In this experiment, as mixing parameter μ increases, MOACO/D-Net can always find partitions with higher NMI values, indicating the effectiveness of MOACO/D-Net.

Next, we executed MOACO/D-Net, CNM and CNM-IC 20 times independently on four real-world networks whose true partitions are unknown. Therefore NMI cannot be obtained for these networks. The comparison of modularity Q values are shown in Fig. 14. For SFI network, the modularity Q obtained by MOACO/D-Net is 0.710 which is slightly smaller than that by CNM. For Netscience network, the Q value got by MOACO/D-Net is 0.910, while the Q value by CNM is 0.955. CNM aims to maximize Q while our MOACO/D-Net is a multi-objective optimization algorithm, which can reveal different community structure. Considering these factors, it can be drawn that MOACO/D-Net is able to deal with large networks with satisfactory results.

4.6 Partitions at different hierarchical levels

In another set of experiments, partitions at different hierarchical levels of the Karate network are presented. Fig. 15 presents the final Pareto Front which consists of 26 nondominated solutions, and the horizontal and vertical axis correspond to the two objective functions shown in Eq. (6). Each solution has a different NMI which is listed near the corresponding vertex. Four solutions (19, 21, 23 and 26) with different number of clusters are chosen. The partition of solution 26 (NMI=0) is drawn as a whole cluster in Fig. 16 (a) and all the vertices in the cluster are drawn in the same color. The partition of two clusters in solution 23 (NMI=1) is drawn in yellow and in orange in Fig. 16 (b). The NMI of solution 23 is 1, which means the partition is the same as the real partition. Fig. 16 (c) shows a result with 3 clusters, which divides the first cluster in Fig. 16 (b) into two clusters. In order to obtain smaller communities, the vertices 25, 26, 29 and 32 can be separated from the second cluster in Fig. 16 (b) and then constitute a new cluster drawn in pink in Fig. 16 (d). It can be seen from Fig. 15 and Fig. 16 that MOACO/D-Net is able to find partitions at different hierarchical levels in a single run, providing more choices for decision makers.

Fig. 17 presents the Pareto Fronts of the Dolphins network, the Football network and the Polbooks network. Four partitions with different number of clusters of the Dolphins network are shown in Fig. 18. It can be observed from Fig. 17 that a group of nondominated solutions can be obtained in one run by our MOACO/D-Net. Each of these solutions corresponds to a different partition of the network consisting of various clusters. The number of clusters is automatically determined by the nondominated solutions resulting from our MOACO/D-Net. Fig. 18 (a) shows the partition of two clusters. The NMI of this partition is 1, indicating it is the same as the real partition. In Fig. 18 (b), (c) and (d), the big cluster in Fig. 18 (a) is further divided into several small clusters by separating several vertices from the original cluster. These different partitions indicate the hierarchical structure of Dolphins network and may provide more useful information for investigators focusing on this network.

Fig. 19 shows four different partitions of the SFI network. SFI network here represents the co-authorship between 118 scientists. The partition with 12 clusters and 0.6955 Q value, the partition with 13 clusters and 0.7057 Q value, the partition with 14 clusters and 0.6925 Q value as well as the partition with 15 clusters and 0.6804 Q value are presented in Fig. 19 (a), (b), (c) and (d) respectively. The true partition of SFI network is not known, but the results shown in Fig. 19 looks interesting and meaningful. Vertices 1, 2, 12 and 24 are always in the center of its own community in different partitions. These vertices have links with almost all the other members within its own community, so it can be inferred that these vertices may be the leaders of these teams. As shown in Fig. 19, some big communities can be further divided into several smaller communities, which is reasonable in real world. Therefore our MOACO/D-Net can reveal different structures of real-world networks that are reasonable, indicating its effectiveness and significance.

5 Conclusions

In this paper, to address the difficulties of applying the multi-objective ACO into the community detection problem including defining and updating the pheromone matrices, constructing the transition probability model, and tuning the parameters, a new transition probability model for the community detection problem is proposed, with which a multi-objective ant colony optimization algorithm based on decomposition (MOACO/D-Net) is designed for finding communities at different hierarchical levels in complex networks. MOACO/D-Net decomposes the community detection problem into a number of single-objective optimization problems, and each subproblem is explored by an ant. All the ants construct solutions according to the proposed transition probability model without the need of knowing the exact number of communities in advance. The solution constructed by an ant is then improved by a simple and effective improvement operator based on the definition of strong community in each generation. Finally a Pareto Front which includes a number of nondominated solutions is obtained, and good partitions at different hierarchical levels can also be obtained in a single run.

Several sets of experiments are designed to find out how the parameters of MOACO/D-Net influence the solutions. Proper ranges of four parameters (pheromone information factor α, heuristic information factor β, persistence rate of pheromone trail ρ and threshold value in solution construction r) in MOACO/D-Net, which are also parameters of single-objective ACO algorithm, are found respectively. For the other two parameters (number of groups K and number of neighbors of each ant T) with no prior knowledge in MOACO/D-Net especially, their influence to the final partition is also given. In our future work, we will focus on improving the efficiency and robustness of the proposed algorithm further and applying it on the overlapping community detection problem.

Acknowledgements

The authors would like to thank the School of Computer Science, University of Nottingham for providing the essential research facilities to this joint research. The authors would also like to thank the anonymous reviewers and editors for their valuable and constructive suggestions, which are helpful for improving our paper. This work was supported by the National Natural Science Foundation of China (No. 61672405), Project supported the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (61621005), the Fundamental Research Funds for the Central Universities (No. JB170204), China Scholarship Council (CSC), the National Natural Science Foundation of China (U1701267, 61473215, 61876141, 61772399, 61773304, 61773300, and 61772393), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), the Major Research Plan of the National Natural Science Foundation of China (Nos. 91438201 and 91438103), and the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT_15R53).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Adriano P, Paolo S (1984) Scalarizing vector optimization problems. Journal of Optimization Theory and Applications 42(4):499-524

Angelini L, Boccaletti S, Marinazzo D, Pellicoro M, Stramaglia S (2007) Identification of network modules by optimization of ratio association. Chaos: An Interdisciplinary Journal of Nonlinear Science, 17(2):023114

Bullnheimer B, Hartl R, Strauss C (1999) An improved ant System algorithm for thevehicle Routing Problem. Annals of Operations Research 89:319-328

Chang H, Feng Z, Ren Z (2013) Community detection using Ant Colony Optimization. IEEE Congress on Evolutionary Computation (pp.3072-3078)

Clauset A, Newman ME, & Moore C. (2004). Finding community structure in very large networks. Physical Review E, 70(2), 066111.

Colorni A, Dorigo M, Maniezzo V, Trubian M (1994) Ant system for job-shop scheduling. Belgian Journal of Operations Research, Statistics and Computer Science 34(1):39-53

Costa D, Hertz A (1997) Ants Can Colour Graphs. Journal of the Operational Research Society 48(3):295-305

Danon L, Diaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment 2005(9):09008

Dorigo M (1992) Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano, Italy

Dorigo M, Gambardella L (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. Evolutionary Computation, IEEE Transactions on, 1(1):53-66

Ehrgott M (2005) Multicriteria Optimization. Springer-Verlag, Berlin, Germany

Eichfelder G (2008) Adaptive Scalarization Methods in Multi-objective Optimization. Springer-Verlag, New York

Fortunato S (2010) Community detection in graphs. Physics Reports 486(3):75-174

Fortunato S, Barthélemy M (2007) Resolution limit in community detection. Proceedings of the National Academy of Sciences of the United States of America, 104(1), 36-41

Fortunato S, Hric D (2016) Community detection in networks: a user guide. Physics Reports, 659, 1-44

Girvan M, Newman M (2002) Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12):7821-7826

Gong M, Cai Q, Chen X, Ma L (2014) Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. Evolutionary Computation, IEEE Transactions on, 18(1):82-97

Gong M, Fu B, Jiao L, Du H. (2011). Memetic algorithm for community detection in networks. Physical Review E 84, 056101.

Gong M, Ma L, Zhang Q, Jiao L (2012) Community detection in networks by using multi-objective evolutionary algorithm with decomposition. Physical review A 391(15):4050-4060

Guédon O, Vershynin R (2016) Community detection in sparse networks via grothendieck's inequality. Probability Theory & Related Fields, 165(3-4), 1-25

Handl J, Knowles J (2007) An evolutionary approach to multi-objective clustering. Evolutionary Computation, IEEE Transactions on, 11(1):56-76

He D, Liu J, Liu D, Jin D, Jia Z (2011) Ant colony optimization for community detection in large-scale complex networks. In Natural Computation (ICNC), 2011 Seventh International Conference on, IEEE, 2:1151-1155

Ji J, Hu R, Zhang H, Liu C (2011) A hybrid method for learning bayesian networks based on ant colony optimization. Applied Soft Computing Journal, 11(4), 3373-3384

Jin D, Liu D, Yang B, Liu J, He D (2011) Ant colony optimization with a new random walk model for community detection in complex networks. Advances in Complex Systems, 14(05):795-815

Ke L, Zhang Q, Battiti R (2013) MOEA/D-ACO: A Multiobjective Evolutionary Algorithm Using Decomposition and Ant Colony. IEEE T. Cybernetics, 43(6):1845-1859

Lancichinetti A, Fortunato S, Kertesz K (2009) Detecting the overlapping and hierarchical community structure of complex networks. New Journal of Physics, 11(3):033015

Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Physical Review E Statistical Nonlinear & Soft Matter Physics,78: 046110.

Li Z, Zhang S, Wang R, Zhang X, Chen L (2008) Quantitative function for community detection. Physical Review E 77(3):036109

Liao T, Stützle T, Oca MAMD, Dorigo M (2014) A unified ant colony optimization algorithm for continuous optimization. European Journal of Operational Research, 234(3):597-609

Lusseau D, Schneider K, Boisseau O, Haase P, Slooten E, Dawson S (2003) The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology, 54(4):396-405

Lyzinski V, Tang M, Athreya A, Park Y, Priebe CE (2017) Community detection and classification in hierarchical stochastic blockmodels. IEEE Transactions on Network Science & Engineering, 4(1), 13-26

Miettinen K (1999) Nonlinear multi-objective optimization. Springer, Vol. 12

Mu C, Liu Y, Liu Y, Wu Jianshe, Jiao Licheng (2014) Two-stage algorithm using influence coefficient for detecting the hierarchical, non-overlapping and overlapping community structure. Physica A Statistical Mechanics & Its Applications,408(408):47-61

Mu C, Zhang J, Jiao L (2014) An Intelligent Ant Colony Optimization for Community Detection in Complex Networks. IEEE Congress on Evolutionary Computation, Beijing, China, pp 700-706

Mu C, Xie J, Liu Y, Chen F, Liu Y, Jiao L (2015) Memetic Algorithm with Simulated Annealing Strategy and Tightness Greedy Optimization for Community Detection in Networks. Applied Soft Computing, 2015, 34: 485-501

Newman M (2011) Communities, modules and large-scale structure in networks. Nature Physics 8(1):25-31

Newman M (2004) Fast algorithm for detecting community structure in networks. Physical review E 69(2):066133

Newman M (2006) Modularity and community structure in networks. Proceedings of the National Academy of Sciences of the United States of America 103(23):8577-8582

Newman M, Girvan M (2004) Finding and evaluating community structure in networks. Physical review E 69(2):026113

Pizzuti C (2008) Ga-net: a genetic algorithm for community detection in social networks. In Parallel Problem Solving from Nature–PPSN X, Springer Berlin Heidelberg, pp 1081

Pizzuti C (2009) A multi-objective genetic algorithm for community detection in networks. Proceedings of the 21st IEEE International Conference on Tools with Artificial Intelligence, Newark, New Jersey, USA, pp 379-386

Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D (2004) Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the United States of America 101(9):2658-2663

Schaub MT, Delvenne JC, Rosvall M, Lambiotte R (2017) The many facets of community detection in complex networks. Applied Network Science, 2(1), 4

Shi C, Wang Y, Wu B, Zhong C (2009) A New Genetic Algorithm for Community Detection. Complex Part II,LNICST 5,pages1298-1309

Stützle T, Hoos H (2000) Max-min ant system. Future Generation Computer Systems 16(8):889-914

Wei Y, Cheng C (1991) Ratio cut partitioning for hierarchical designs. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 10(7):911-921

Zachary W (1977) An information flow model for conflict and fission in small groups. Journal of anthropological research 33(4):452-473

Zhang AY, Zhou HH (2016) Minimax rates of community detection in stochastic block models. Computer Science, 44(5)

Zhang Q, Li H (2007) MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. Evolutionary Computation, IEEE Transactions on, 11(6):712-731

Zhou HF, Li J, Li JH, Zhang FC, Cui YA (2017) A graph clustering method for community detection in complex networks. Physica A Statistical Mechanics & Its Applications, 469, 551-562

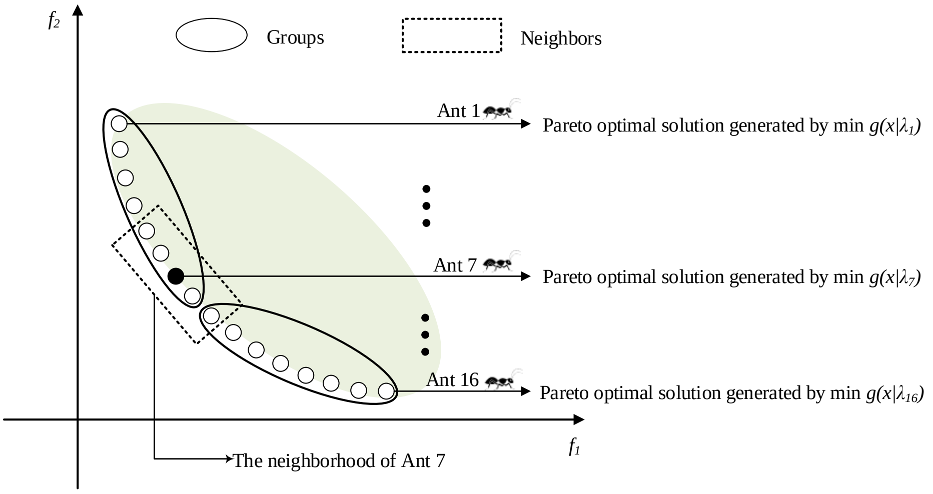
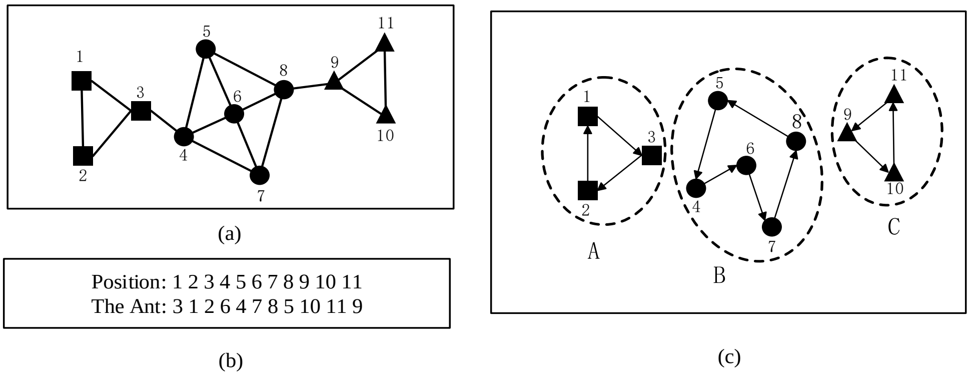

f2
Groups
Neighbors
Ant 1
Pareto optimal solution generated by min g(x|λ1)
Ant 7
Pareto optimal solution generated by min g(x|λ7)
Ant 16
Pareto optimal solution generated by min g(x|λ16)
f1
The neighborhood of Ant 7

Fig. 1 An example of multi-objective optimization problem partitioned into 16 subproblems, each explored by an ant, respectively. Ant i in the Pareto Front is a Pareto optimal solution defined by min $g(x|\lambda_i)$. All the ants in the same ellipse belong to the same group, i.e. group 1 includes ants 1, 2,…, 8 and group 2 includes ants 9, 10, …, 16. Ant 5, 6, 8 and 9 in the rectangle are neighbors of ant 7.



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

Fig. 2 An illustrative example of the locus-based adjacency scheme. (a) An example network with 11 vertices and 16 edges (b) A possible encoded solution (3 1 2 6 4 7 8 5 10 11 9), where the allele value of the 1st element is 3, and the allele value of the 2nd element is 1, etc. (c) The corresponding decoded solution, where the network is partitioned into three clusters: A, B and C (the graph is shown as directed to aid understanding on how it originates from the encoded solution).

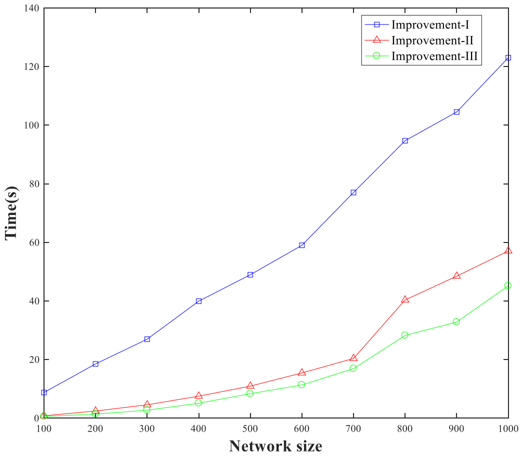


Fig. 3 The relationship between the run-time of three improvement operators and the network size on LFR benchmark networks

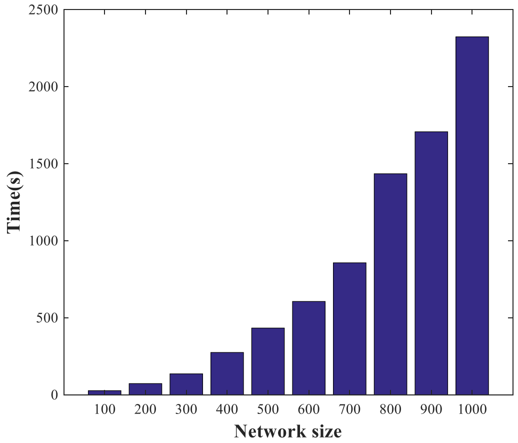


Fig. 4 The relationship between the run-time of MOACO/D-Net-III and the network size on LFR benchmark networks

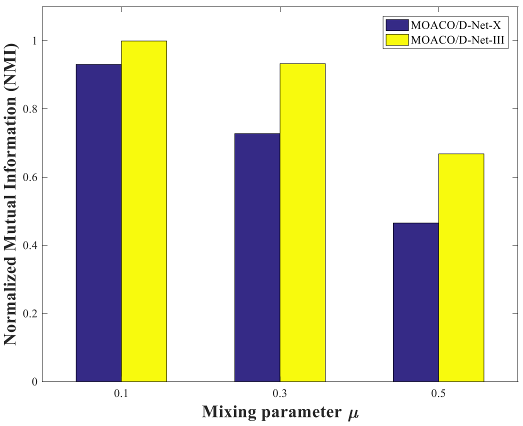
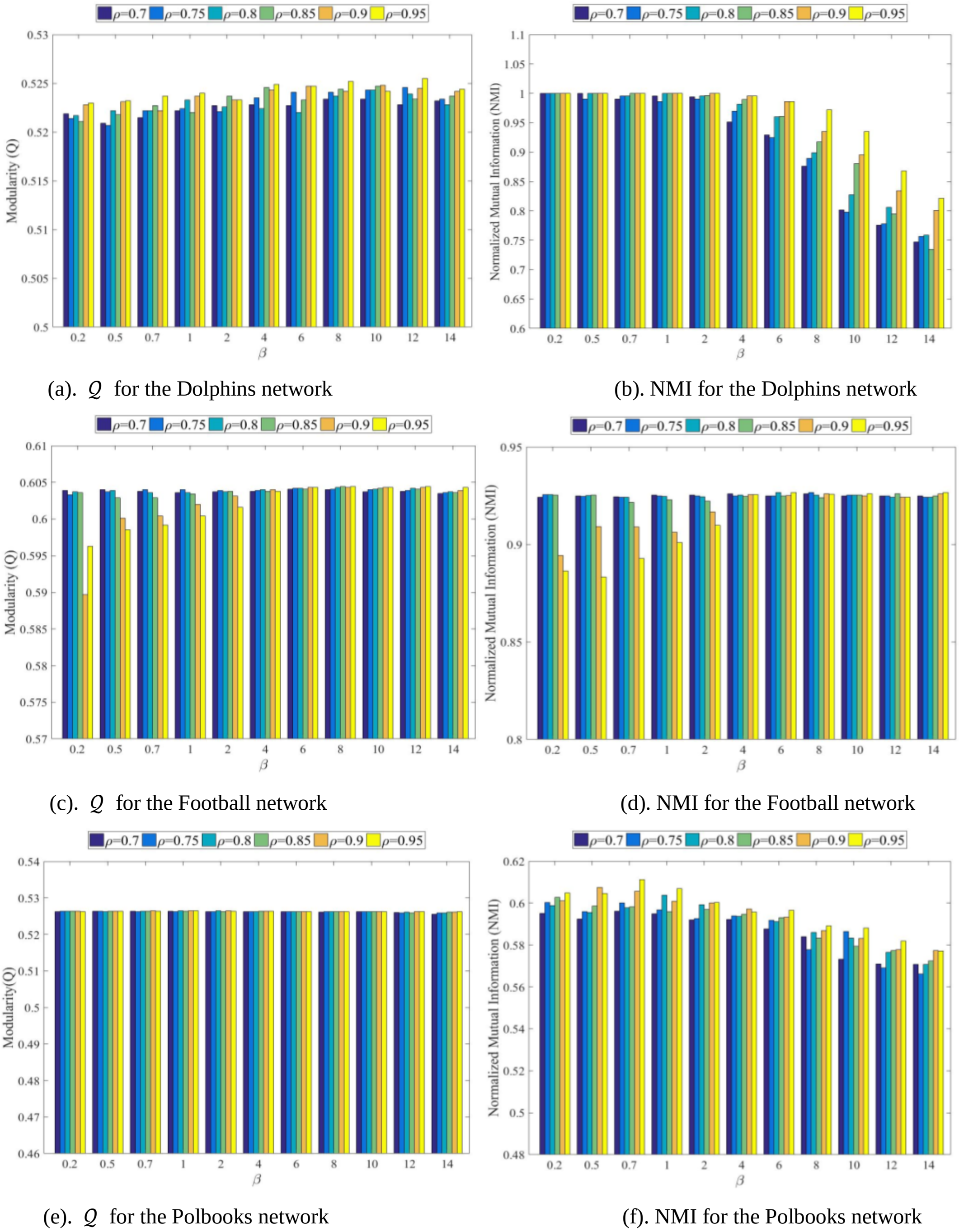


Fig. 5 Comparison of MOACO/D-Net-X and MOACO/D-Net-III on LFR benchmark networks with different mixing parameters



(a). Q for the Dolphins network

(b). NMI for the Dolphins network

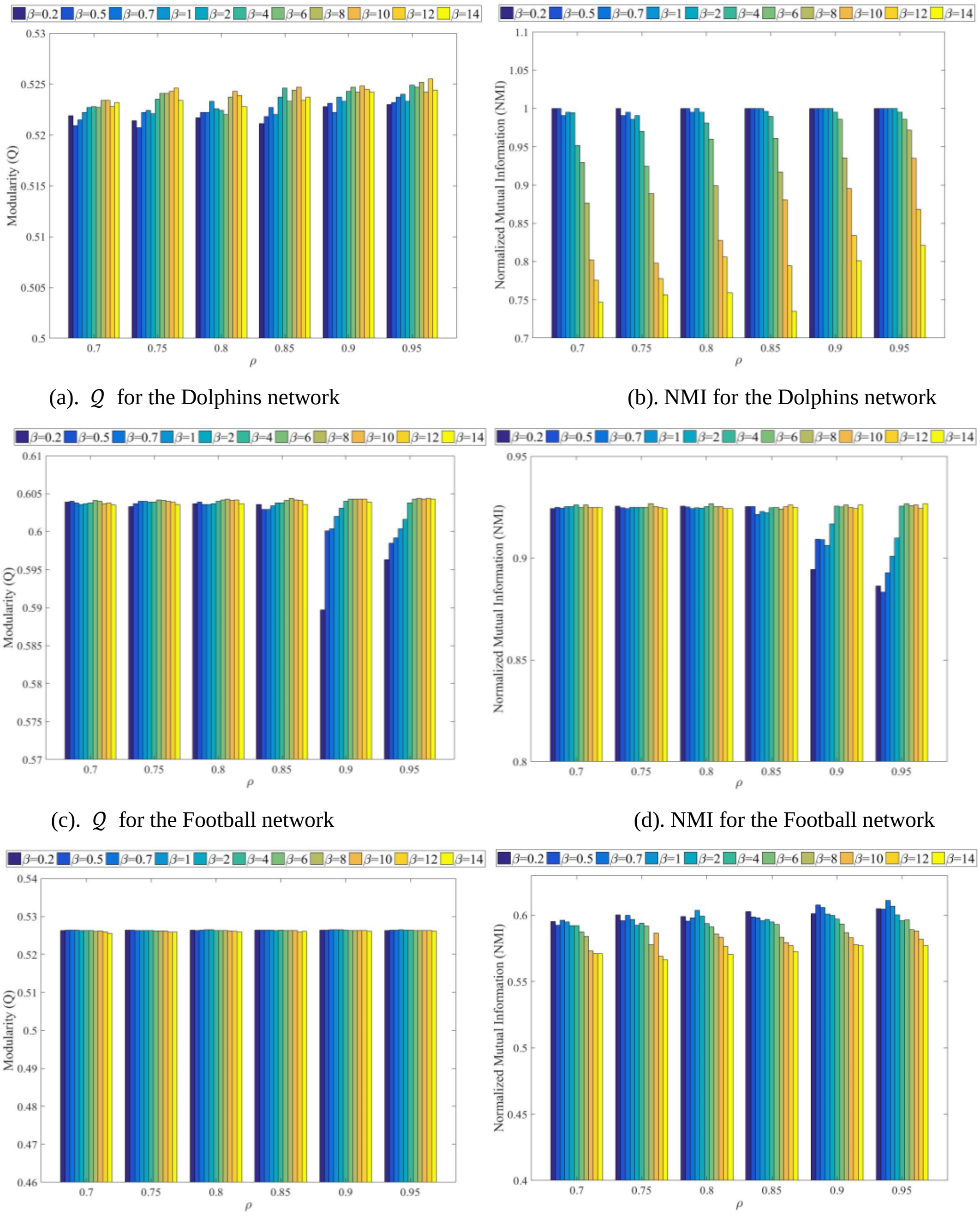
(c). Q for the Football network

(d). NMI for the Football network

(e). Q for the Polbooks network

(f). NMI for the Polbooks network

Fig. 6 Results of comparison experiments on the value of β (when α=1 and ρ=0.7, 0.75, 0.8, 0.85, 0.9, 0.95) for real-world networks.



(a). Q for the Dolphins network

(b). NMI for the Dolphins network

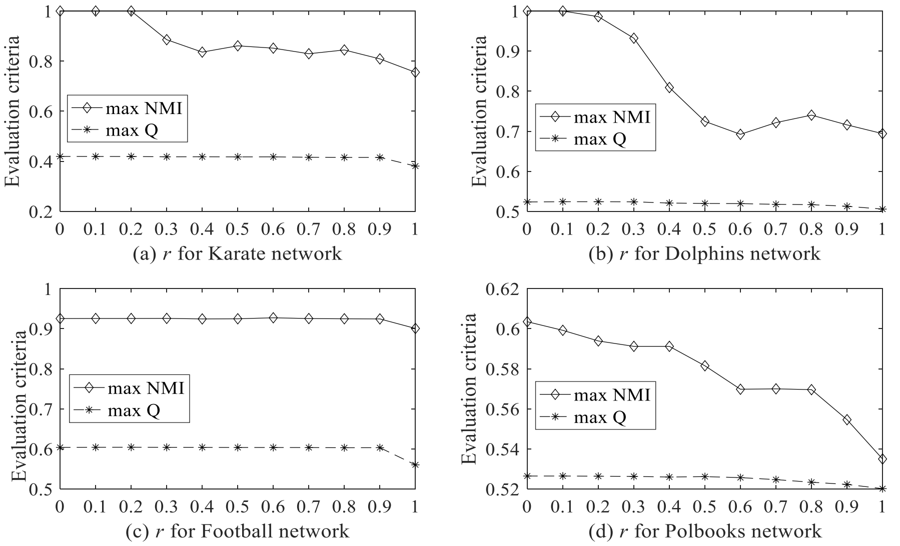
(c). Q for the Football network

(d). NMI for the Football network

(e). Q for the Polbooks network

(f). NMI for the Polbooks network

Fig. 7 Results of comparison experiments on the value of ρ (when α=1 and β=0.2，0.5，0.7，1, 2, 4, 6, 8, 12, 14) for real-world networks.



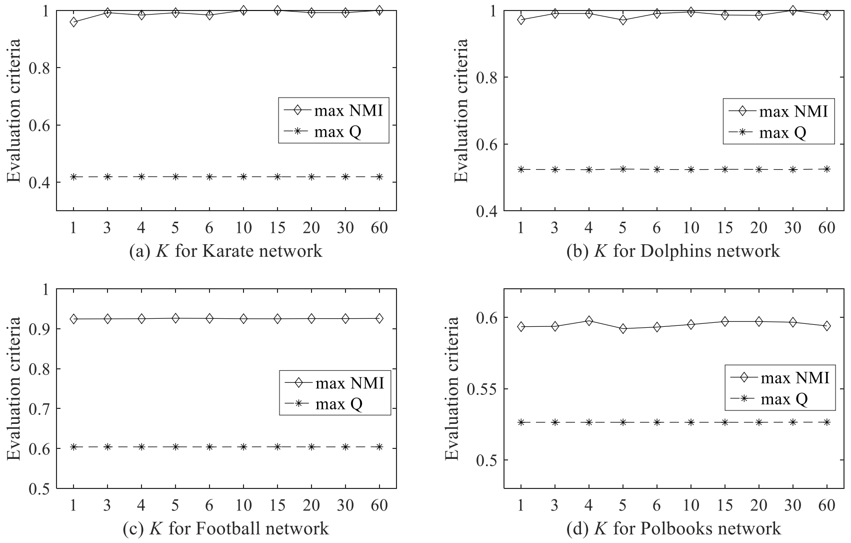
(a) *r* for Karate network

(b) *r* for Dolphins network

(c) *r* for Football network

(d) *r* for Polbooks network

Fig. 8 Plots of the max NMI and the max Q obtained with different values of r for four real-world networks, where $r = \{0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1\}$.



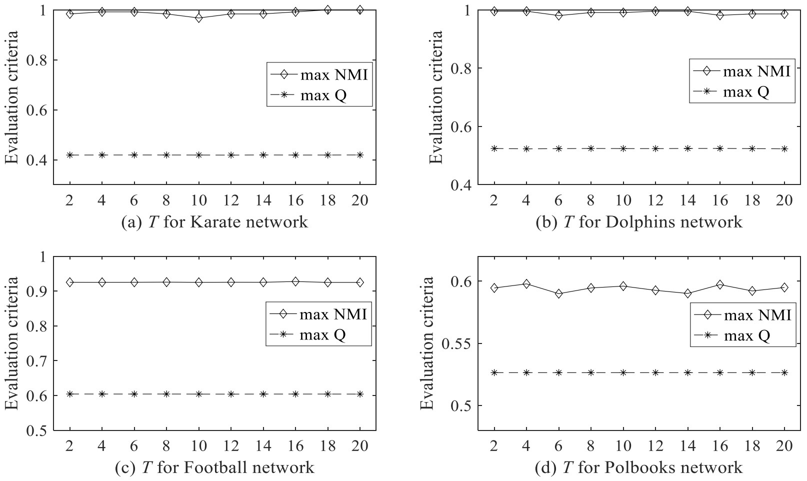
(a) *K* for Karate network

(b) *K* for Dolphins network

(c) *K* for Football network

(d) *K* for Polbooks network

Fig. 9 Plots of the max NMI and the max Q obtained with different values of *K* for four real-world networks, where $K = \{1,3,4,5,6,10,15,50,30,60\}$.



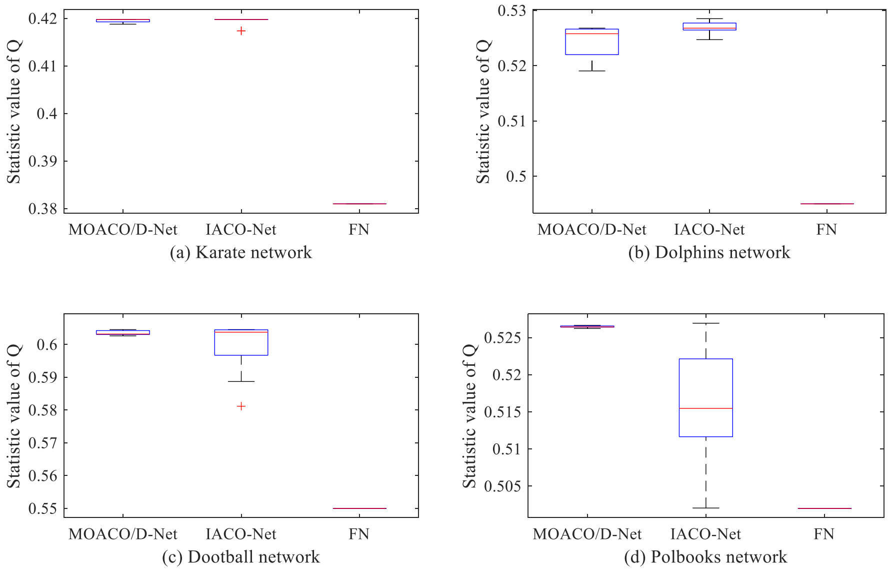
(a) T for Karate network

(b) T for Dolphins network

(c) T for Football network

(d) T for Polbooks network

Fig. 10 Plots of the max NMI and Q obtained with different values of T for four real-world networks, where $T = \{2,4,6,8,10,12,14,16,18,20\}$.



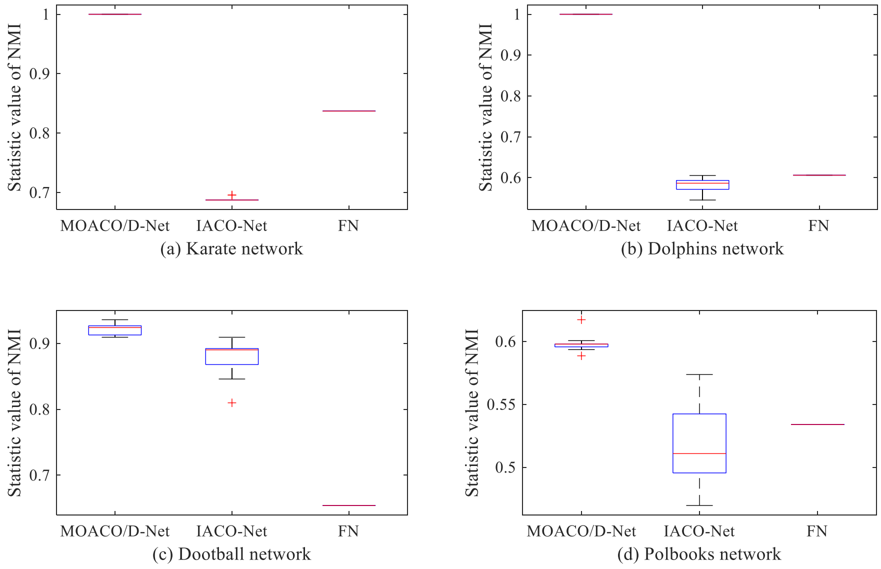
(a) Karate network

(b) Dolphins network

(c) Dootball network

(d) Polbooks network

Fig. 11 The box plot of the statistic value of Q obtained by MOEA/D-Net, IACO-Net and FN over the 20 runs on the real-world networks.



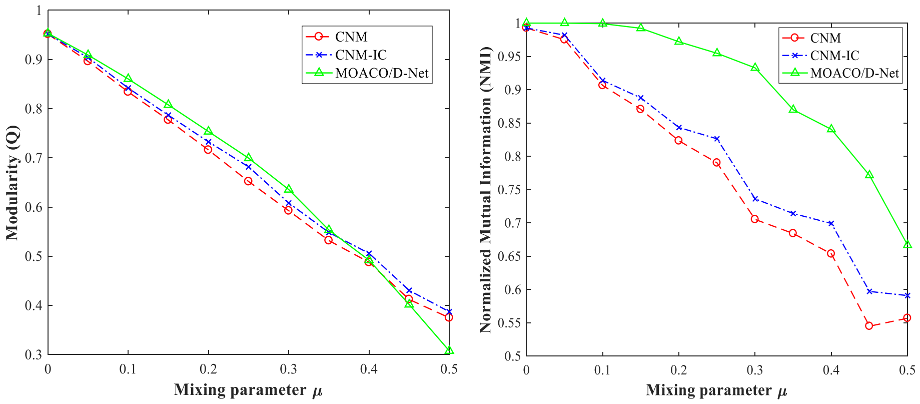
(a) Karate network

(b) Dolphins network

(c) Dootball network

(d) Polbooks network

Fig. 12 The box plot of the statistic value of NMI obtained by MOEA/D-Net, IACO-Net and FN over the 20 runs on the real-world networks.



(a) Q values

(b) NMI

Fig. 13 Average Q values and NMI of different algorithms on LFR benchmark networks with 1000 nodes

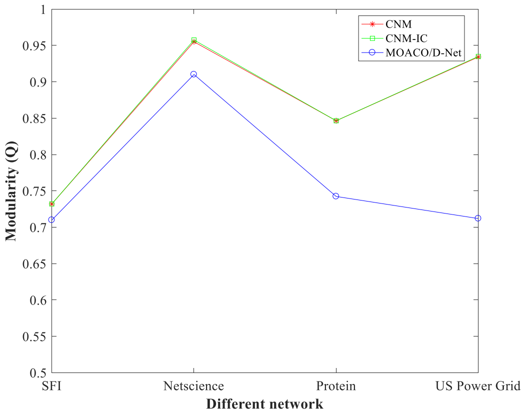


Fig. 14 Average Q values of different algorithms on real-world networks with real partitions unknown

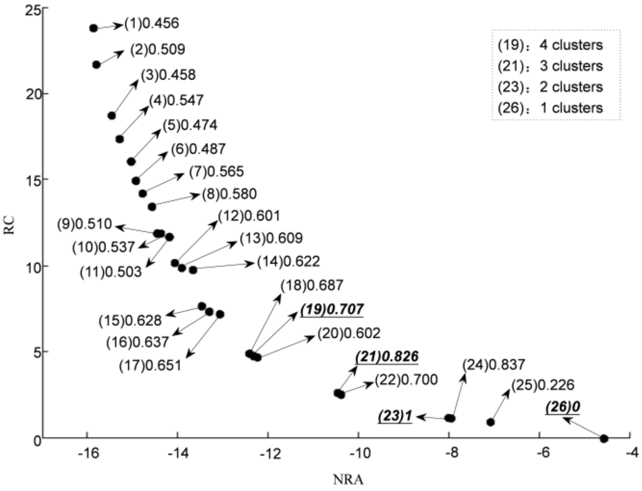
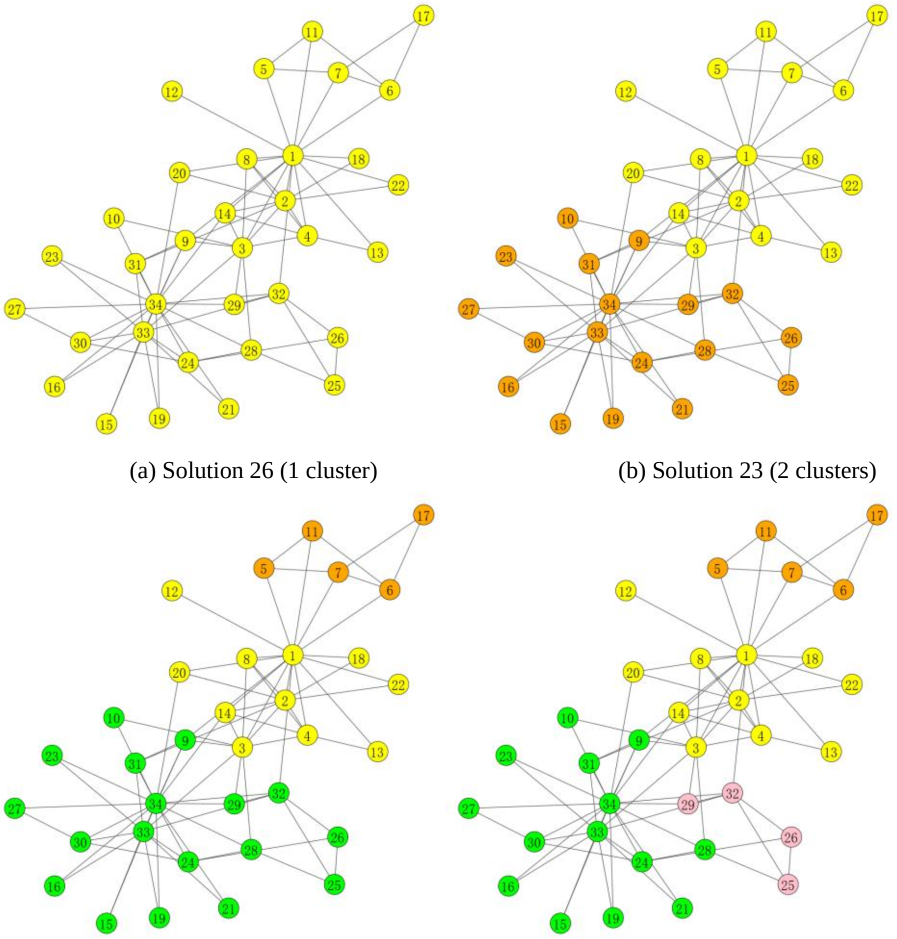


Fig. 15 Pareto Front of the Karate network with 26 nondominated solutions obtained by MOEA/D-Net. The NMI of each solution is shown.



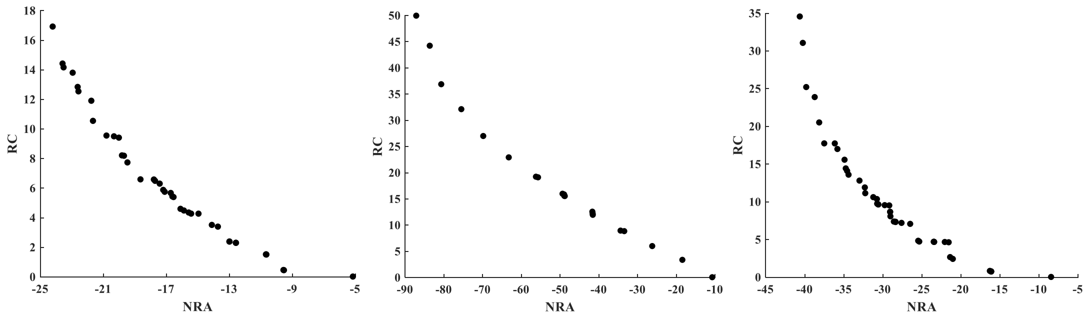
(a) Solution 26 (1 cluster)

(b) Solution 23 (2 clusters)

(c) Solution 21 (3 clusters)

(d) Solution 19 (4 clusters)

Fig. 16 Four different partitions for the Karate network: one cluster in (a), two clusters in (b), three clusters in (c) and four clusters in (d). Vertices in different clusters are drawn with different colors

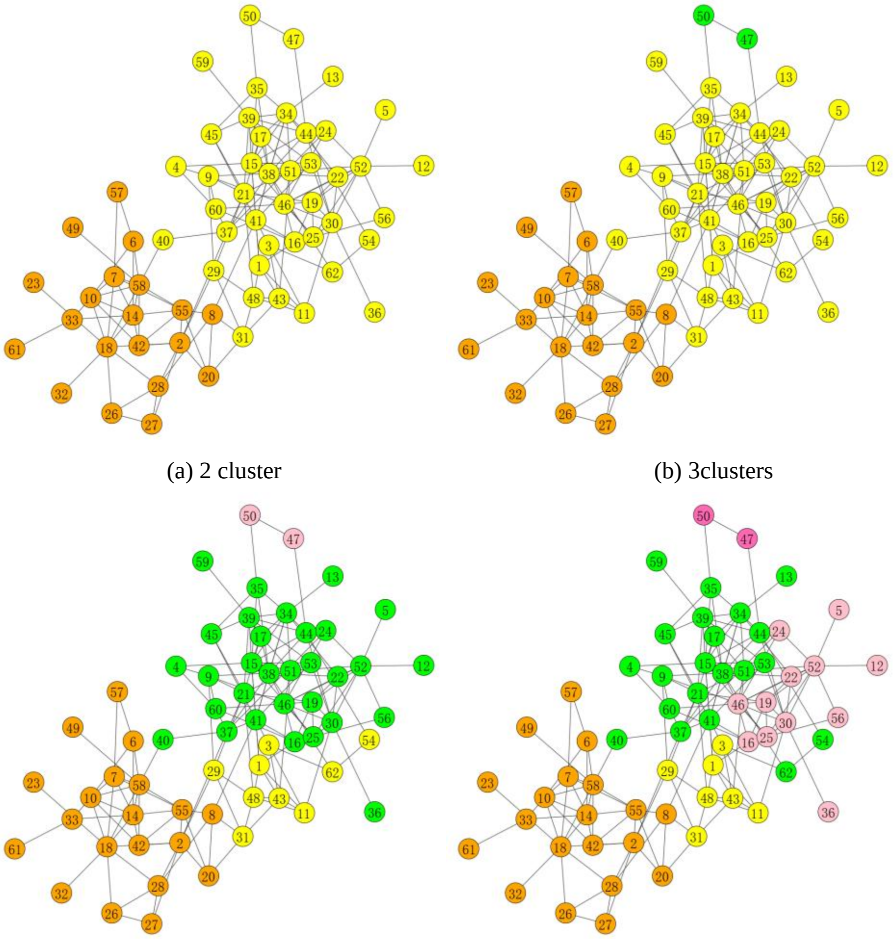


(a) Dolphins network

(b) Football network

(c) Polbooks network

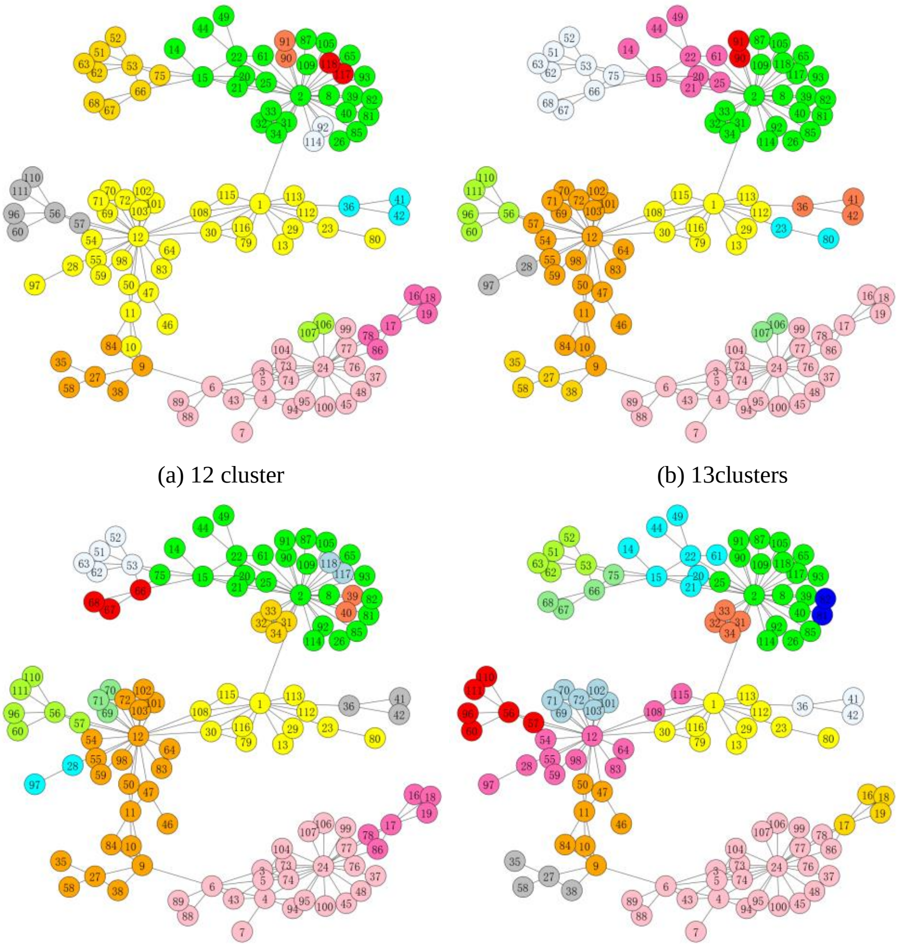
Fig. 17 Pareto Fronts obtained by MOEA/D-Net on the Dolphins network, the Football network and the Polbooks



(a) 2 cluster (b) 3clusters

(c) 4 cluster (d) 5clusters

Fig. 18 Four different partitions for the Dolphins network: two cluster in (a), three clusters in (b), four clusters in (c) and five clusters in (d). Vertices in different clusters are drawn with different colors



(a) 12 cluster

(b) 13clusters

(c) 14 cluster

(d) 15 clusters

Fig. 19 Four different partitions for the SFI network: 12 cluster in (a), 13 clusters in (b), 14 clusters in (c) and 15 clusters in (d). Vertices in different clusters are drawn with different colors