Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition

Maoguo Gong, *Member, IEEE,* Qing Cai, Xiaowei Chen, and Lijia Ma

***Abstract*—The field of complex network clustering has been very active in the past several years. In this paper, a discrete framework of the particle swarm optimization algorithm is proposed. Based on the proposed discrete framework, a multiobjective discrete particle swarm optimization algorithm is proposed to solve the network clustering problem. The decomposition mechanism is adopted. A problem-specific population initialization method based on label propagation and a turbulence operator are introduced. In the proposed method, two evaluation objectives termed as kernel k-means and ratio cut are to be minimized. However, the two objectives can only be used to handle unsigned networks. In order to deal with signed networks, they have been extended to the signed version. The clustering performances of the proposed algorithm have been validated on signed networks and unsigned networks. Extensive experimental studies compared with ten state-of-the-art approaches prove that the proposed algorithm is effective and promising.**

***Index Terms*—Clustering, complex networks, evolutionary algorithm, multiobjective optimization, particle swarm optimization.**

I. Introduction

A. Complex Network Clustering

RECENT years have witnessed an enormous interest in complex networks, since many complex systems, including collaboration networks [1], the World Wide Web [2], [3], and neural networks [4], can be modeled as complex networks. The task of network clustering is to partition a complex network into several groups, which hold the conditions that, for an unsigned network, connections between the nodes in the same group are dense and connections between different groups are sparse; for a signed network, links both within and between groups are dense. Such a group is called a community, a cluster, or a module [5]. Network clustering is essential for understanding how a network is organized and also to understand its functions. A number of network clustering techniques have been developed over the last decade. A recent survey can be found in [5].

Naturally, network clustering can be modeled as an optimization problem [6]. Very often, such a problem is NP-hard. For this reason, evolutionary algorithms (EAs) have been used for network clustering. For example, Pizzuti [7] has proposed a single objective genetic algorithm (GA-net) for network clustering, and Gong *et al.* [8] have suggested a memetic algorithm-based network clustering method (Meme-net). In some applications, it is desirable to consider several conflicting objectives at the same time in network clustering. Therefore, multiobjective evolutionary algorithms can also find their niche in this area. Several multiobjective evolutionary network clustering methods, such as MOGA-net [9], MOCD [10], and MOEA/D-net [11], have recently been proposed.

B. Multiobjective Evolutionary Algorithm and Particle Swarm Optimization

Multiobjective evolutionary algorithms (MOEAs) aim at producing a set of Pareto optimal solutions to a multiobjective optimization problem in a single run. Due to their wide applications, much effort has been devoted to MOEAs and a number of MOEAs have been developed. Most current MOEAs can be classified into three categories. The first one is Pareto dominance based; the well-known algorithms in this category are NSGA-II [12], SPEA2 [13], PAES [14], and AMOSA [15]. The second category is performance indicator based; the hypervolume is the most popular indicator used in these algorithms. The third category is decomposition based; these algorithms decompose a multiobjective optimization problem (MOP) into a number of single objective subproblems or simple multiobjective subproblems, and use a population search method to solve these subproblems in a collaborate way.

Particle swarm optimization (PSO), originally proposed for single objective continuous optimization problems, is a population based optimization method [16]. PSO works with a population (i.e., swarm) of particles. Particles move in the search space to find optimal solutions. A particle adjusts its velocity before its movement according to some simple rules. These rules, inspired by the movement of a bird flock or fish school, make use of the best position visited by each particle and the global best solution produced by the swarm to drive particles to a promising region.

Manuscript received November 15, 2012; revised March 7, 2013; accepted April 20, 2013. Date of publication April 30, 2013; date of current version January 27, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61273317, the National Top Youth Talents Support Program of China, and the Fundamental Research Fund for the Central Universities under Grants K50510020001 and K5051202053.

The authors are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, Shaanxi Province 710071, China (e-mail: gong@ieee.org; 506183509@qq.com; xiaowei7002@126.com; 723816009@qq.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2013.2260862

Besides its simplicity, a major advantage of PSO over classic optimization methods is that it makes very few assumption about the problem to solve. It does not require the problem to have very good mathematical properties. PSO can be applied to various difficult optimization problems such as noisy and dynamic problems [17]–[19]. PSO has been proved to be one of the most popular optimization techniques. Ever since its emergence, it has found nationwide applications in many domains [20]–[25], among which is the MOEA. Over the past decade, many PSO-based MOEAs (MOPSOs) have been proposed. Table I summarizes the typical MOPSOs that emerged in the literature.

Due to the success of PSO in continuous optimization, some attempts have been made to design discrete PSO (DPSO) algorithms. The first DPSO was the binary PSO (BPSO) algorithm proposed by Kennedy and Eberhart in [38]. However, the binary coding scheme confounds its applications. Intuitively, as long as we can map the continuous space into a discrete one (e.g., rounding technique), continuous PSO can be easily turned into DPSO. Based on this idea, some DPSO algorithms characterized by the space transformation technique have been put forward [39], [40]. By redefining the position and the velocity of a particle, Clerc [41] outlined a swap-operator-based DPSO algorithm in which the position of a particle is defined as a permutation of numbers and the velocity as a set of swaps. Some scholars define the position and velocity as a fuzzy matrix [42], [43] and convert position into a feasible solution. Recently, Chen *et al.* in [44] have advanced a set-based DPSO (S-PSO) in which the candidate solution and velocity are defined as a crisp set, and a set with possibilities, respectively. All arithmetic operators in the velocity and position updating rules used in the original PSO are replaced by the operators and procedures defined on crisp sets, and sets with possibilities in S-PSO. The S-PSO method proves to be promising for combinatorial optimization problems.

C. Contributions of This Paper

1) For the complex network clustering problem, a discrete PSO framework is proposed. Based on network topology, we redefine the particle's velocity and position vectors in discrete form. We also redefine all the arithmetical operators between velocity and position vectors. A problem-specific particle swarm initialization approach and a turbulence operator are introduced.
2) Based on the proposed discrete PSO framework, a discrete MOPSO algorithm referred to as MODPSO is proposed. In the proposed MODPSO algorithm, a decomposition mechanism is adopted. To the best of our knowledge, the approach proposed in this paper represents the first attempt for applying a methodology based on discrete MOPSO to the domain of complex network clustering.
3) The objective functions used in MODPSO have been extended to the signed version. MODPSO minimizes two objectives termed as kernel k-means and ratio cut. But their neglect of the signed features of networks confound their applications. In order to handle signed networks, the two objectives have been extended to the signed version. Extensive experiments on signed networks and unsigned networks validate the good performances of the proposed algorithm.

The rest of this paper is organized as follows. Section II gives the problem statements and our motivations. In Section III, the proposed method for network clustering is presented. In Section IV, the performances of the proposed algorithm are validated on both signed networks and unsigned networks, we also compare our algorithm with ten state-of-the-art approaches. The conclusion is finally summarized in Section V.

II. Problem Statements and Motivations

A. Network Community Definition

A network with both positive and negative links is called a signed network [45]. An unsigned network can be regarded as a special case of a signed network, i.e., there are only positive links. Radicchi *et al.* [46] gave a qualitative unsigned network community definition based on node degree. Given an unsigned network denoted as $G = (V, E)$, where V and E are the aggregations of vertices and links, respectively. Let k_i be the degree (the number of links that have connections with node i) of node i and A be the adjacency matrix of G. $A_{ij} = 1$, if there is a link between node i and j, otherwise, $A_{ij} = 0$. Given that $S \subset G$ is a subgraph, where node i belongs to, let $k_i^{in} = \sum_{i,j \in S} A_{ij}$ and $k_i^{out} = \sum_{i \in S, j \notin S} A_{ij}$ be the internal and external degree of node i, then S is a community in a strong sense if

$$\forall i \in S, \quad k_i^{in} > k_i^{out}. \tag{1}$$

S is a community in a weak sense if

$$\sum_{i \in S} k_i^{in} > \sum_{i \in S} k_i^{out}. \tag{2}$$

Signed network communities are defined not only by the density of links but also by the signs of links. Given is a signed network modeled as $G = (V, PE, NE)$, where V is the set of nodes and PE and NE are the set of positive and negative links, respectively. Let A be the weighted adjacency matrix of G and e_{ij} be the link between node i and j. Then the element of A is defined as: $A_{ij} = 1, if\ e_{ij} \in PE$; $A_{ij} = -1, if\ e_{ij} \in NE$; $A_{ij} = 0, if \nexists e_{ij}$. Given that $S \subset G$ is a subgraph where node i belongs to, let $(k_i^+)^{in} = \sum_{j \in S, e_{ij} \in PE} A_{ij}$ and $(k_i^-)^{in} = \sum_{j \in S, e_{ij} \in NE} |A_{ij}|$ be the positive and negative internal degrees of node i, respectively. Then S is a community in a strong sense if

$$\forall i \in S, \quad (k_i^+)^{in} > (k_i^-)^{in}. \tag{3}$$

Let $(k_i^-)^{out} = \sum_{j \notin S, e_{ij} \in NE} |A_{ij}|$ and $(k_i^+)^{out} = \sum_{j \notin S, e_{ij} \in PE} A_{ij}$ be the negative and positive external degree of node i, respectively. S is a community in a weak sense if

$$\begin{cases} \sum_{i \in S} (k_i^+)^{in} > \sum_{i \in S} (k_i^+)^{out} \\ \sum_{i \in S} (k_i^-)^{out} > \sum_{i \in S} (k_i^-)^{in} \end{cases}. \tag{4}$$

Thus, in a strong sense, a node has more positive links than negative links within the community; in a weak sense,

TABLE I
REPRESENTATIVE MOPSO ALGORITHMS IN THE LITERATURE

Year	Author	Key Features	Reference
2002	Parsopoulos and Vrahatis	aggregating functions, scheme similar to lexicographic ordering	[26]
2002	Coello Coello et al.	global repository, used a turbulence operator	[27]
2003	Hu et al.	used secondary population	[28]
2003	Mostaghim and Teich	used sigma method to find best local guide	[29]
2003	Zhang et al.	selected the *gbest* as the average of the complete set of the *gbest*	[30]
2004	Baumgartner et al.	linear aggregating functions, adopted a gradient technique	[31]
2004	Pulido and Coello Coello	Pareto dominance, various clustering techniques	[25]
2005	Srinivasan and Seow	hybridized PSO with EA, self updating mechanism	[32]
2006	Köppen and Veenhuis	fuzzy Pareto dominance	[**?**]
2009	Benameur et al.	fuzzy clustering technique	[34]
2011	Li et al.	attraction and repulsion mechanism	[35]
2011	Daneshyari and Yen	cultural framework	[36]
2012	Moubayed et al.	dominance concept, decomposition approach	[37]

the positive links within a community are dense while the negative links between different communities are also dense.

B. Network Clustering Problem

Network clustering problem can be formulated as an optimization problem. For this purpose, Girvan and Newman [47] proposed the concept of modularity (normally, denoted as Q) that is written as

$$Q = \frac{1}{2m}\sum_{i,j}(A_{ij} - k_i k_j/2m)\delta(i,\, j) \tag{5}$$

where m is the size of edges; A_{ij} is the element of the adjacent matrix of the network; k_i is the degree of node i; and $\delta(i,\, j) = 1$, if node i and j are in the same community, otherwise, 0.

Q is designed for unsigned networks. For signed networks, Gómez *et al.* [48] presented a reformulation of Q that allows the analysis of signed networks. The signed Q is formulized as

$$SQ = \frac{1}{2w^+ + 2w^-}\sum_{i,j}(w_{ij} - (\frac{w_i^+ w_j^+}{2w^+} - \frac{w_i^- w_j^-}{2w^-}))\delta(i,\, j) \tag{6}$$

where w_{ij} is the weight of the signed adjacency matrix and $w_i^+(w_i^-)$ denotes the sum of all positive (negative) weights of node i.

The foundation of modularity is based on the idea that the existence of communities is revealed by the comparison between the actual density of links in a subgraph and the density one would expect to have in the subgraph if the nodes of the graph were connected regardless of community structures. Normally, by assumption we take it that the larger the value of Q, the stronger the community structure is.

However, Fortunato and Barthlemy have pointed out the resolution limitation of modularity in [49] that modularity optimization may not identify communities smaller than a certain scale, which depends on the total size of the network and the degree of interconnectedness of the modules. The signed modularity also cannot avoid this limitation. To overcome this problem, Li *et al.* [50] have introduced a quality function, called modularity density. The authors have introduced a tunable parameter to the objective function equation that allows one to explore the network at different scales. Lancichinetti *et al.* [51] have introduced the concept of community fitness (CF) and Pizzuti [7] has put forward the community score (CS) criteria to stop the division of the network. Other measurements based on modularity have also been established [52]–[54].

In our previous work in [11], we formulated the unsigned network clustering problem as a multiobjective optimization problem, in which two objectives termed as negative ratio association (NRA) and ratio cut (RC) were to be minimized. These two objectives can also break through the modularity limitation. Given an unsigned network denoted as $G = (V, E)$ with $|V| = n$ vertices and $|E| = m$ edges. The adjacency matrix is A. Provided that a partition with k clusters of the network is $\Omega = \{c_1, c_2, ..., c_k\}$, say $V_1, V_2 \in \Omega$, we define $L(V_1, V_2) = \sum_{i\in V_1, j\in V_2} A_{ij}$ and $L(V_1, \overline{V_2}) = \sum_{i\in V_1, j\in \overline{V_2}} A_{ij}$, where $\overline{V}_2 = \Omega - V_2$. Then the optimization problem can be defined as

$$\min \begin{cases} NRA = -\sum_{i=1}^{k} \frac{L(V_i, V_i)}{|V_i|} \\ RC = \sum_{i=1}^{k} \frac{L(V_i, \overline{V_i})}{|V_i|} \end{cases}. \tag{7}$$

In this paper, we have changed the NRA into the kernel k-means (KKM) introduced in [55]; thus, we reformulate the unsigned network clustering problem as

$$\min \begin{cases} KKM = 2(n-k) - \sum_{i=1}^{k} \frac{L(V_i, V_i)}{|V_i|} \\ RC = \sum_{i=1}^{k} \frac{L(V_i, \overline{V_i})}{|V_i|} \end{cases}. \tag{8}$$

The motivations to define the above objectives are that, as it has been pointed out in [55], that the KKM is a decreasing function of the number of communities while the opposite trend happens to the RC function. In other words, they are two conflicting objectives. From the definitions of KKM and RC, we can notice that the right operand of KKM can be considered as the sum of the density of the link of intracommunities. RC can be considered as the sum of the density of the link of intercommunities. To minimize KKM and RC, we can ensure that the links within a community are dense while the links between communities are spare and this is in accordance with the basic nature of the communities in an unsigned network.

The signed network clustering problem is a little different from that of the unsigned one. In order to handle signed

networks, we amend our objective functions KKM and RC in the signed version. Consequently, we reformulate the signed network clustering problem as the following optimization problem:

$$\min \begin{cases} SRA = -\sum_{i=1}^{k} \frac{L^+(V_i,V_i)-L^-(V_i,V_i)}{|V_i|} \\ SRC = \sum_{i=1}^{k} \frac{L^+(V_i,\overline{V_i})-L^-(V_i,\overline{V_i})}{|V_i|} \end{cases} \tag{9}$$

where $L^+(V_i, V_j) = \sum_{i\in V_i, j\in V_j} A_{ij}, (A_{ij} > 0)$ and $L^-(V_i, V_j) = \sum_{i\in V_i, j\in V_j} |A_{ij}|, (A_{ij} < 0)$.

To minimize SRA and SRC we can make sure that the positive links within a community are dense while the negative links between communities are also dense, which is in accordance with the feature of signed community.

C. Our Motivations

In this paper, by redefining the particle's velocity and position and the arithmetical operators between them, a DPSO framework for solving complex network clustering problem is proposed for the first time. Based on the proposed DPSO framework, a discrete MOPSO algorithm based on decomposition is proposed. Problem-specific population initialization and turbulence operator are introduced to promote diversity. Considering the signed features of complex networks, the objective functions have been extended into signed version.

1) *Motivations for Choosing PSO Framework for Complex Network Clustering:* The high speed of convergence of the algorithm, the simple framework, the easy implementation of the method, and a large amount of existing PSO variants algorithms in public domain make PSO feasible and promising for solving some optimization problems.

Traditional clustering algorithms, such as K-Means and the spectral clustering approach, need to know the number of clusters in advance, which affects the clustering performances. For the network clustering problem, it is very inconvenient if the size of the network is large, because the number of the clusters may be very big. However, our proposed algorithm can automatically determine the clusters. This point will be further illustrated in Section IV.

A complex network has a lot of prior knowledge, such as the linkage relations and the degree information, which should be taken into consideration. In this paper, during the designing of our algorithm, we make full use of this prior knowledge. Besides, from the perspective of new solution generation, traditional EAs normally generate new solutions through genetic operators that seldom consider any knowledge acquired in the former generations, while on the contrary, PSO takes advantage of personal best and global best information to generate new solutions. All this convinces us that choosing PSO would be more proper.

2) *Motivations for Proposing the Discrete MOPSO Algorithm:* To overcome the modularity resolution limitation, methods such as those in [50] and [52]–[54] have been established; however, the shared common disadvantage for these methods is that they all need an extra parameter that is hard to tune. In this paper, by simultaneously optimizing two conflicting objective functions termed as kernel k-means and ratio cut, we can overcome this drawback as well as require no tuning parameter. This is the very motivation for us to adopt the MOPSO to solve network clustering problem.

Most, if not all, of the existing MOPSOs are mainly designed for continuous optimization problems and we find it hard to extend these existing algorithms to the network clustering problem. In order to solve this problem, in this paper, we have proposed a discrete MOPSO algorithm based on network topology.

3) *Motivations for Introduced Mechanisms to Preserve Diversity:* In our algorithm, we adopt the decomposition method used in MOEA/D [56] to promote diversity. Our algorithm decomposes the MOP into scalar aggregation problems. Decomposition transforms the MOP into a set of distinct scalar aggregation problems. Every particle solves the corresponding problem by applying priorities to each objective according to its weighting vector. This assists the optimization process to find potential solutions that are evenly distributed along the Pareto front and to mitigate premature convergence.

Problem-specific initialization is an important issue in EA application. In our algorithm, the population initialization method based on label propagation is adopted to generate individuals with high clustering efficiency.

Furthermore, a problem-specific turbulence operator is designed. The turbulence operator makes full use of network linkage information. To obtain a tradeoff between the search ability and the convergence, the turbulence is higher in the initial stage to aid exploration, and then gradually decreases to promote exploitation, and at last becomes zero to ensure convergence.

III. Description of the Proposed Method

In this section, the proposed MODPSO method for network clustering is described. First, a particle representation scheme and its updating rules used in the proposed DPSO framework are given, and then the swarm initialization and leader selection operation are described. The framework of MODPSO and the turbulence operator are elaborated, and at last the complexity of MODPSO is discussed.

A. Definition of Discrete Position and Velocity

In order to solve the complex network clustering problem, in this paper, we redefine the term position and velocity used in PSO in discrete form. The definitions are as follows:

1) *Definition of position*: In PSO, the position vector represents a solution to the optimized problem. For the network clustering problem, the position permutation of a particle i is defined as $X_i = \{x_1, x_2, ..., x_n\}$. Each dimension of position is a random integer between 1 and n, i.e., $x_i \in [1, n]$, where n is equal to the total vertices number of the network. If $x_i = x_j$, then we take it that node i and j belong to the same cluster.

Fig. 1 gives an illustration of how the discrete position of a particle is coded and decoded.

The motive behind the definition of the position vector is that it is straightforward and easy to decode, so that it will lower the computational complexity.

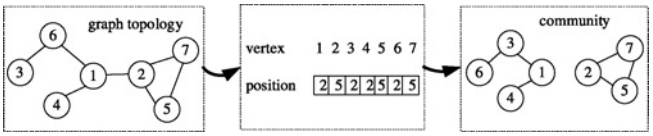


Fig. 1. Generic illustration of particle representation.

2) *Definition of velocity*: Velocity works on the position sequence and it is rather crucial. A good velocity gives the particle an guidance and determines whether the particle can reach its destination and by how fast it could. The discrete velocity of particle i is defined as $V_i = \{v_1, v_2, ..., v_n\}$. V_i is binary-coded, and if $v_i = 1$, the corresponding element x_i in the position vector will be changed, otherwise, x_i keeps its original state.

The first motivation of the velocity definition, in the above style, is to prevent particles from flying away because, in general, it is necessary to set a threshold V_{max} to inhibit particles from flying out of the boundaries. But since our velocity is binary-coded, we no longer need V_{max} parameter. The second motivation lies in the very definition of position. The defined position vector is integer coded, how to define a proper velocity to work on the position is nontrivial. Our defined velocity actually reflects the differences between two position vectors.

B. Discrete Particle Status Updating

In the proposed DPSO, a velocity provides a particle with the moving direction and tendency. After updating the velocity, one particle makes use of the new velocity to build new position. Since the position and velocity, in our approach, are all integer vectors, the mathematical updating rules in continuous PSO no longer fit the discrete situation; therefore, we redefine them to meet the requirements of network clustering problem. We first redefine the velocity updating rule in discrete form as

$$V_i = sig(\omega V_i + c_1 r_1 (Pbest_i \oplus X_i) + c_2 r_2 (Gbest \oplus X_i)) \quad (10)$$

where ω is the inertia weight; c_1 and c_2 are the cognitive and social components, respectively; and r_1 and r_2 are two random numbers with range [0,1].

In (10), $\oplus$ is defined as a XOR operator and the function $Y = sig(X)$, where $Y = (y_1, y_2, ..., y_n)$, $X = (x_1, x_2, ..., x_n)$, is defined as

$$\begin{cases} y_i = 1 & if \quad rand(0,1) < sigmoid(x_i) \\ y_i = 0 & if \quad rand(0,1) \geq sigmoid(x_i) \end{cases} \quad (11)$$

where the sigmoid function is defined as

$$sigmoid(x) = \frac{1}{1 + e^{-x}}. \quad (12)$$

In our algorithm, to promote exploration and exploitation, the inertia weight ω is randomly generated between [0,1], and the cognitive and social components c_1 and c_2 are set to the typical value of 1.494.

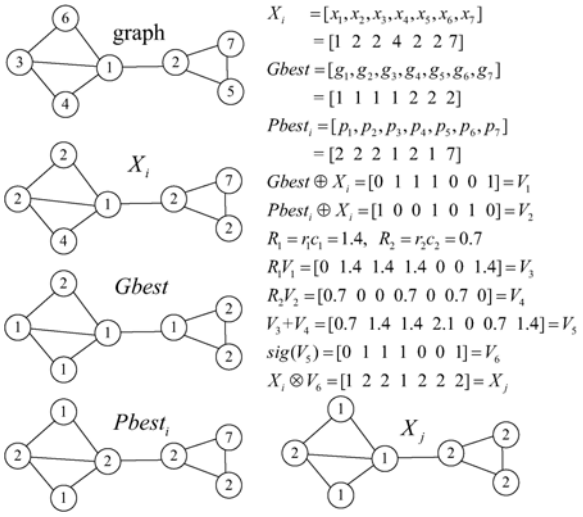


Fig. 2. Schematic example of how a particle updates its status.

Based on the newly defined discrete velocity updating rule, we now redefine the position updating rule as the following discrete form:

$$x_i^t = x_i^t \otimes v_i^t. \quad (13)$$

In (13), the operator $\otimes$ is the key procedure in the particle status updating process. It directly affects the performance of the algorithm since the right operand of $\otimes$ determines the direction to which the particle flies. A good operator $\otimes$ should help to guide the particle to a better place that is much closer to the food rather than a bad position that is far away from it.

Given a position $X_1 = \{x_{11}, x_{12}, ..., x_{1n}\}$ and a velocity $V = \{v_1, v_2, ..., v_n\}$, position$\otimes$velocity generates a new position that corresponds to a new solution to the optimization problem, i.e., $X_1 \otimes V = X_2$, $X_2 = \{x_{21}, x_{22}, ..., x_{2n}\}$. The element of X_2 is defined as

$$\begin{cases} x_{2i} = x_{1i} & if \quad v_i = 0 \\ x_{2i} = Nbest_i & if \quad v_i = 1 \end{cases} \quad (14)$$

where $Nbest_i$ is an integer. Suppose vertex i has a neighbor set $N = \{n_1, n_2, ..., n_k\}$, then $Nbest_i$ is calculated by

$$Nbest_i = \arg\max_r \sum_{j \in N} \varphi(x_{1j}, r) \quad (15)$$

where $\varphi(i, j) = 1$, if $i = j$, otherwise, 0. The function $\arg\max_r f(x)$ returns the value of r that maximizes $f(x)$. So, in (15), $Nbest_i$ actually equals to the label identifier possessed by the majority of the neighbors of node i. To calculate $Nbest_i$ in this way makes sense, because in reality it is more possible for one member to join the community that is formed by the majority of its friends.

A schematic example of the detailed operations about the discrete particle status updating rules can be found in Fig. 2. In Fig. 2, X_i and $Pbest_i$ are the current position and the personal best position of particle i, respectively. $Gbest$ is the global best solution of the swarm. V_1, V_2, V_3, V_4, and V_5 are the intermediate variables. V_6 and X_j are figured out by (10) and (13), respectively.

TABLE II
COMPARISON BETWEEN DIFFERENT LEADER SELECTION MECHANISMS

Strategy	Time Complexity	Diversity	Impact on Dimension
random	$O(N)$	good	slight
nearest	$O(DAN)$	normal	middle
sigma	$O((D(D-1)/2)AN)$	bad	great
stripe	$O(N)$	good	two dimensions

D is the objective dimension, A is the archive size and $N$is the particle swarm size.

From what is illustrated above we see that the proposed DPSO framework has the following features.

1) The definitions of discrete position and velocity are straight forward and very simple.
2) The newly defined arithmetic operators are very easy to realize, which greatly lower down the computational complexity.
3) The proposed DPSO framework does not need to know the clusters of a network in advance; it can automatically determine it by itself.

The newly designed DPSO framework seems to be very suitable for solving network clustering problem.

C. Particle Swarm Initialization

A good initialization mechanism may both reduce the searching space so as to save the time for the algorithm to reach global optima and promote diversity. A traditional random initialization method generates initial solutions with a lot of redundance. In this paper, a label propagation-based initialization strategy proposed in our previous work is introduced. As the labels propagate, densely connected groups of nodes quickly reach a consensus on a unique label. For more information please refer to our early work in [57].

D. Selection of Leaders

In a MOPSO, diversity can be promoted through the selection of leaders. Several strategies for choosing a *gbest* as the leader for each particle including the random method that randomly chooses a member from the archive, the nearest method that chooses the archive member that is nearest to the particle in objective space, the sigma approach suggested by Mostaghim *et al.* [29] and the stripe strategy advanced by Villalobos-Arias *et al.* [58] are compared in Table II.

In this paper, we choose the random strategy because it can promote good population diversity as well as having small time complexity and what is most important is that it has a slight impact on the objective dimensions since the dimensions of our objectives are rather high. For one particle, there are *ns* corresponding neighbors (defined based on the Euclidean distances between the aggregation weight coefficient vectors); then we randomly choose one particle from the neighbors as the leader to guide the flight.

E. Framework of the Proposed Algorithm

In the proposed algorithm, the adopted decomposition method is the widely used Tchebycheff approach, which is written as

$$g^{te}(x|w, z^*) = \max_{1\le i\le k} w_i|f_i(x) - z_i^*| \quad \text{subject to } x \in \Omega \tag{16}$$

where $z^* = (z_1^*, z_2^*, \cdots, z_k^*)$ is the reference point, i.e., $z_i^* = \{\min f_i(x)|x \in \Omega\}$ for each $i = 1, ..., k$.

In our algorithm, the Tchebycheff approach is used. Because both of the two objectives used in this paper are not continuous, we cannot simply conclude that is concave or not. If it is nonconcave, the weighted sum approach [56] could not work well. This is why we prefer the Tchebycheff approach.

The whole framework of the proposed MODPSO algorithm for complex network clustering is given in Algorithm III-E. In Step 3.8, the *pbest* is updated by using the concept of Pareto dominance, i.e., if the newly generated solution dominates the *pbest* solution, then updates it with the new solution; if *pbest* dominates the newly produced solution, *pbest* remains in its original state; if they are mutually nondominated, then we make use of aggregation method to determine whether to update *pbest* or not. The process is given in Algorithm III-E.

Algorithm 1 Framework of the proposed MODPSO.

Parameters: max generation: *maxgen,* number of decomposed sub-problems: *ns*, swarm size: *pop*, mutation probability *pm*, inertia weight: ω, the learning factors: c_1, c_2. **Input**: The adjacent matrix A of a network. **Output**: Pareto front solutions. Each solution corresponds to a partition of a network.

Step 1) Initialization
Step 1.1) Position initialization: $P = \{x_1, x_2, ..., x_{pop}\}^T$.
Step 1.2) Velocity initialization: $V = \{v_1, v_2, ..., v_{pop}\}^T$.
Step 1.3) Generate a well-distributed weighted vectors: $W = \{w_1, w_2, ..., w_{pop}\}^T$.
Step 1.4) Personal best position initialization: $Pbest = \{pbest_1, pbest_2, ..., pbest_{pop}\}^T$, $pbest_i = x_i$.
Step 1.5) Initialize reference point z^*.
Step 1.6) Initialize neighborhood N based on Euclidean distance, i.e., $N = \{n_1, n_2, ..., n_{pop}\}^T$.
Step 2) set $t = 0$. // the number of flight cycles
Step 3) Cycling
for $i = 1, 2, ..., pop$, **do**
Step 3.1) Randomly select one particle from the neighbors as the *gbest* particle, i.e., $gbest \leftarrow random(x_i.neighbor)$.
Step 3.2) Calculate new velocity v_i^{t+1} for the ith particle according to Eq. 10.
Step 3.3) Calculate new position x_i^{t+1} for the ith particle according to Eq. 13.
Step 3.4) If $t < maxgen \cdot pm$, Turbulence operation on x_i^{t+1}, see *Algorithm* III-F for more information.
Step 3.5) Evaluation of x_i^{t+1}.
Step 3.6) Update neighborhood solutions: for the jth ($j = 1, 2, ..., N$) particle in the neighborhood of the ith particle, if $g^{te}(x_i^{t+1}|w_j, z^*) \le g^{te}(x_i^t|w_j, z^*)$, then $x_i^t = x_i^{t+1}$, $F(x_i^t) = F(x_i^{t+1})$.
Step 3.7) Update reference point z^*.
Step 3.8) Update personal best solution $pbest_i$.
Step 4) Stopping criteria: If $t < maxgen$, then $t++$ and go to **Step 3**, otherwise, stop the algorithm and output.

Algorithm 2 *pbest* updating process.

```
if x_i^{t+1} ≺ pbest_i
    pbest_i = x_i^{t+1}
else
    if x_i^{t+1} ≻ pbest_i
        do nothing;
    else
        if w_{i1} f(x_i^{t+1}) + w_{i2} f(x_i^{t+1}) < w_{i1} f(pbest_i) + w_{i2} f(pbest_i)
            pbest_i = x_i^{t+1};
        end if
    end if
end if
```

Algorithm 3 Pseudo code of turbulence operation on one particle.

```
for i = 0; i < vertex; i++
    if rand(0, 1) < pm
        for j = 0; j < node[i].neighborsize; j++
            x[node[i].neighbor[j]] = x[i];
        end for
    end for
end for
```

F. Turbulence Operator

To preserve diversity and help a MOPSO to escape from local optima, many existing MOPSOs adopt the turbulence operator. For the network clustering problem, in our algorithm, the adopted turbulence operator is called a neighbor based mutation (NBM). The procedure can be depicted as follows. First we generate a pseudo random number between 0 and 1; for each gene in every chromosome, if the random number is smaller than the mutation probability *pm*, the NBM process is applied to the gene, namely, assigns its label identifier to all of its neighbors. The pseudo code is given in Algorithm III-F.

G. Complexity Analysis

1) *Space Complexity*: In our algorithm, two main memorizers are needed. The first one is the clustering data memorizer, which needs a complexity of $O(n^2)$, and n is the number of vertices of the network. The second memorizer is for the particles, say there are N particles, then the complexity is $O(Nn)$. Thus, the total space complexity of our algorithm is $O(n^2)$.
2) *Computational Complexity*: The main time complexity lies in Step 3 of our algorithm since Step 1 can be accomplished in linear time. Here, we use n and m to denote the vertex and edge numbers of the network, respectively. Step 3.1 and 3.7 need $O(1)$ basic operation. Step 3.2, 3.6, and 3.8 need $O(n)$ basic operations, and Step 3.3 requires $O(D^2)$ basic operations where D is the averaged degree of a network. Step 3.4 needs $O(n)$ basic operations. Step 3.5 needs $O(m+n)$ basic operations. So the worst case time complexity is $4O(n)+O(D^2)+O(m+n)+2O(1)$. According to the operational rules of the symbol O, the worst case time complexity for MODPSO can be simplified as $O(pop \cdot maxgen \cdot (m+n))$, where *pop* and *maxgen* are the population size and iteration number, respectively.

TABLE III
PARAMETERS OF THE ALGORITHMS

Algorithm	*pop*	*maxgen*	*pc*	*pm*	*ns*	Reference
MODPSO (A1)	100	100		0.1	40	[ours]
MOPSO-r1 (A2)	100	100		0.1		[25]
MOPSO-r2 (A3)	100	100		0.1		[59]
GA-net (A4)	100	100	0.9	0.1		[7]
Meme-net (A5)	100	100	0.9	0.1		[8]
MOGA-net (A6)	100	100	0.9	0.1		[9]
MOCD (A7)	100	100	0.9	0.1		[10]
MOEA/D-net (A8)	100	100	0.9	0.1	40	[11]
GN (A9)						[47]
CNM (A10)						[60]
Informap (A11)		100				[61]

pop Represents the population size, *maxgen* denotes the max iterations of the algorithm, *pc* and *pm* are the crossover and mutation possibility, respectively, and *ns* is the neighborhood size.

IV. Experimental Studies

A. Comparison Algorithms

In this paper, five EA-based algorithms named as GA-net, Meme-net, MOGA-net, MOCD, MOEA/D-net, and two MOPSO-based algorithms referred to as MOPSO-r1 and MOPSO-r2 are chosen to compare with the proposed approach. We also compare our method with several other well-known avenues named as GN, CNM, and Informap. The experimental parameters of the algorithms are listed in Table III.

The MOPSO algorithm proposed in [25] uses the concept of Pareto dominance to allow the heuristic to handle problems. It employs an external repository of particles that is used by other particles to guide their own flight. However, this algorithm is originally designed for continuous MOPs. In order to make it possible for comparison, based on the original framework of MOPSO in [25], the definitions of particles (position and velocity) and their mutual operations are made the same as that in our algorithm. Thus, a revised MOPSO (denoted as MOPSO-r1) for network clustering is introduced as an comparison algorithm. In [59], Palermo *et al.* put forward a DPSO algorithm for multiobjective design space exploration by making use of aggregation technique to extend the formulation of the PSO to the multiobjective domain. A revised version of this algorithm for network clustering is denoted as MOPSO-r2. In MOPSO-r1 and MOPSO-r2, the turbulence operators are the same as ours.

GA-net was proposed by Pizzuti. It provided a feasible scheme to combine EA with network clustering problem. The author defined an evaluation index called community score to check the performance of the algorithm. We choose it as our comparison algorithm because we want to show that an EA-based algorithm is likely to become involved in local optima.

Meme-net was put forward in our previous paper. In this paper, a new EA-based framework was introduced to reveal community structures in networks. We also proposed a local

search strategy based on a greedy mechanism to speed up the algorithm convergence. We chose it as our comparison algorithm because we want to show that the proposed MODPSO can perform better even without local search mechanism.

MOCD, MOGA-net, and MOEA/D-net are three MOEA-based network clustering algorithms. In order to make a comparison between the MOEA-based and the MOPSO-based algorithms, these three algorithms are chosen as the comparison approaches. In our algorithm, the decomposition strategy is utilized, and so is MOEA/D-net. The main difference between the two approaches is the optimization strategy. MOEA/D net uses a genetic algorithm to optimize subproblems while our algorithm makes use of the proposed DPSO.

GN is a divisive hierarchical clustering algorithm proposed by Girvan and Newman. In this algorithm, the authors have first brought forward a well-known and widely traced network partitioning evaluation index, i.e., the modularity. The performance of GN is remarkably good though it is computationally complicated.

CNM was proposed by Clauset, Newman, and Moore [60]. This method is essentially a fast implementation of the GN approach. The code of CNM is available at .

Informap was proposed by Rosvall and Bergstrom [61]. This algorithm is based on the information theory. It uses the probability flow of random walks on a network as a proxy for information flows in the real system and decomposes the network into modules by compressing a description of the probability flow.

B. Experimental Settings

The proposed algorithm is written in C++. The experiments have been performed on a Inter(R) Celeron(R)M CPU 520 machine, 1.6 GHz, 512 MB memory. The operating system is MS Windows XP and the compiler is VC++ 6.0. The C++ code of our algorithm is available by request to the first author through e-mail.

For the case when the ground truth of a network is known, we adopt the so called normalized mutual information (NMI) index described in [62] to estimate the similarity between the true clustering results and the detected ones. Given two partitions A and B of a network, let C be the confusion matrix whose element C_{ij} is the number of nodes shared in common by community i in partition A and by community j in partition B. The NMI(A,B) is then defined as

$$NMI = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{C_A} C_{i.}log(C_{i.}/N) + \sum_{j=1}^{C_B} C_{.j}log(C_{.j}/N)} \quad (17)$$

where $C_A(C_B)$ is the number of clusters in partition $A(B)$, $C_{i.}(C_{.j})$ is the sum of elements of C in row i(column j), and N is the number of nodes of the network. If $A = B$, then NMI(A,B)=1; if A and B are totally different, then NMI(A,B)=0. The NMI is a similarity measure proved to be reliable in [63].

For the case when the ground truth of a network is unknown, we use the modularity Q and the signed modularity SQ as the evaluation indexes. We also make use of these indexes to select the ultimate clustering solution from the Pareto front.

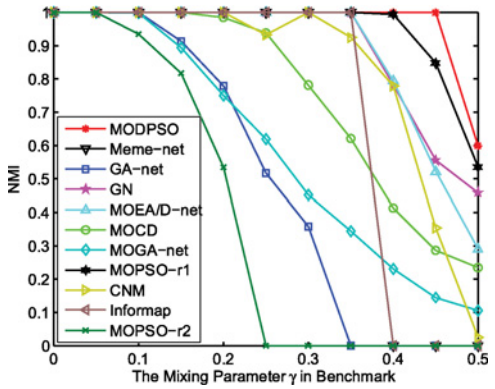


Fig. 3. Max NMI values averaged over 30 runs for different algorithms on GN extended benchmark datasets.

C. Experiments on Unsigned Networks

1) *Experiments on GN Extended Benchmark Networks:* Methods to detect communities in graphs need to be thoroughly tested. So, we first do some experiments on the extended benchmark network with a built-in community structure proposed by Lancichinetti *et al.* in [64], which is an extension of the classical GN benchmark proposed by Girvan and Newan in [47]. The extended benchmark network consists of 128 nodes divided into four communities of 32 nodes each. Every node has an average degree of 16. It shares a fraction $1-\gamma$ of links with the rest in its community, and γ with the other nodes of the network. Here, γ is called the mixing parameter. When $\gamma < 0.5$, the neighbors of a vertex inside its community are more than the neighbors belonging to the rest groups; in other words, the network has strong community structure. On the contrary, the structure is rather vague. We test all the algorithms on 11 computer-generated networks with the values of γ ranging from 0.0 to 0.5.

Fig. 3 summarizes the maximum NMI values averaged over 30 runs for different algorithms when the mixing parameter γ increases from 0.0 to 0.5 with interval 0.05. As is shown in Fig. 3, when the mixing parameter is no bigger than 0.35, almost all the methods, except GA-net, MOGA-net, MOPSO-r2, and MOCD, can figure out the true partitions (NMI equals 1). As the mixing parameter increases, the community structure in the network is becoming fuzzy gradually. It becomes harder and harder to figure out the true structures. When $\gamma = 0.4$, our algorithm gets $NMI = 1$ while MOPSO-r1 gets $NMI = 0.9952$. But when $\gamma = 0.45$, our algorithm still gets $NMI = 1$; our algorithm is still able to correctly classify the fuzzy network. From the curves we can see that the proposed MODPSO algorithm is more powerful than the rest; though the MOPSO-r1 approach is just a little shy of ours, it is much slower since it keeps on updating the external archive, which is time consuming.

In our proposed algorithm, two diversity promotion mechanisms are introduced, i.e., label propagation-based initialization (LPI) and turbulence operator. In order to show their impacts on the performance of the whole algorithm, some experiments are done here. In terms of the particle swarm initialization, here, we compare the introduced label propagation with random initialization and heuristic initialization introduced in our previous work in [8]. Then, we run our

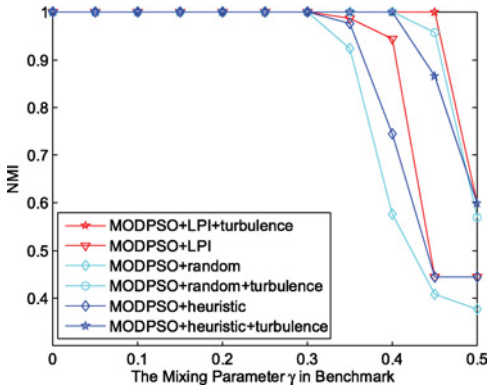


Fig. 4. Max NMI values averaged over 30 runs for MODPSO with arbitrary combinations between the initialization methods and turbulence operator.

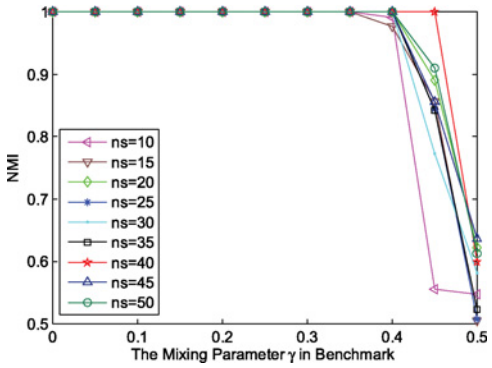


Fig. 5. Max NMI values averaged over 30 runs for MODPSO with different neighborhood size.

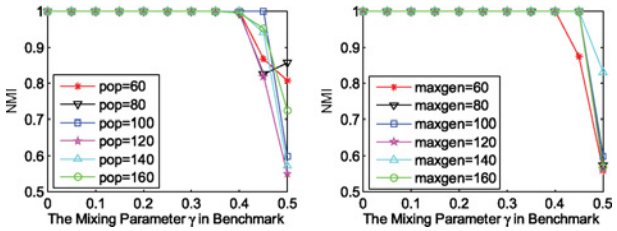


Fig. 6. Max NMI values averaged over 30 runs for MODPSO with different population size and iteration numbers.

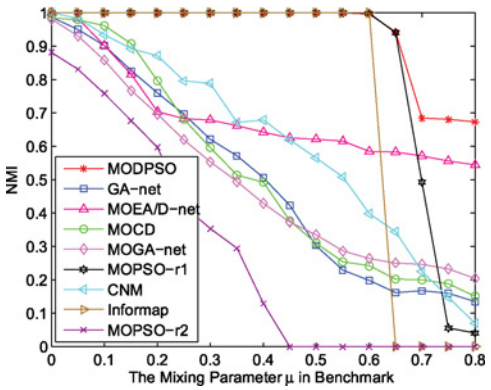


Fig. 7. Max NMI values averaged over 30 runs for different algorithms on LFR benchmark datasets.

algorithm with the arbitrary combinations between the initialization methods and turbulence operator. The statistical results are shown in Fig. 4. It is clearly shown that the introduced two mechanisms can greatly improve the performance of the algorithm, especially the turbulence operator.

In our algorithm the parameter *ns* is used to determine the neighborhood size. *ns* should be neither too small nor too large. To test its influence on the performance of our algorithm, we run the algorithm 30 times with different *ns*; the averaged results are shown in Fig. 5. From the results we see that $ns = 40$ seems perfect for our algorithm. Thus, in our later experiments, *ns* is fixed as 40.

Apart from the parameter *ns*, the population size *pop* and the iteration number *maxgen* parameters are also crucial. To test their influence on the performances of the algorithm, we fix one of them as 100, then we let the other one change from 60 to 160 with interval 20. Fig. 6 shows the statistical *NMI* values obtained by different *pop* and *maxgen* numbers.

We can observe from the curves that when *maxgen* is fixed as 100, $pop = 100$ seems perfect and when *pop* is fixed as 100, $maxgen = 140$ performs the best and $maxgen = 80, 100, 120, 160$ perform similarly well. Considering both the computational complexity and the algorithm performance, in our algorithm, *pop* and *maxgen* are all set as 100.

2) *Experiments on LFR Benchmark Networks:* The above benchmark networks do not reflect some important features in graph representations of real systems, such as the fat-tailed distributions of node degree and community size, because on those benchmark networks, all vertices have approximately the same degree. Moreover, all communities have exactly the same size by construction. Therefore, new classes of benchmark graphs called LFR have been proposed by Lancichinetti [64], in which the distributions of node degree and community size are both power laws with tunable exponents. They assume that the distributions of degree and community size are power laws, with exponents τ_1 and τ_2, respectively. Each vertex shares a fraction $1 - \mu$ of its edges with the other vertices of its community and a fraction μ with the vertices of the other communities; $0 \leq \mu \leq 1$ is the mixing parameter. The software to create the LFR benchmark graphs can be freely downloaded at . In our experiments, we generate 17 networks with the mixing parameter increasing from 0 to 0.8 with an interval of 0.05. Each network contains 1000 nodes and the cluster size ranges from 10 to 50. $\tau_1 = 2$ and $\tau_2 = 1$; the averaged degree for each node is 20 and the max node degree is 50. We run every algorithm 30 times and the statistical results are exhibited in Fig. 7.

From this figure we see that, when μ is bigger than 0.1, only three algorithms can correctly classify these networks, but with the growth of μ, when it surpasses 0.6, none of these three approaches can make it. However, from the curves we can draw the conclusion that our proposed algorithm performs remarkably well and it is superior to the rest.

3) *Experiments on Real-world Networks:* We now show the applications of MODPSO on six real-world networks, i.e., the Zachary's karate club network [65], the dolphin social network [66], the American college football network [47], the Santa Fe Institute (SFI) network [47], the netscience network

TABLE IV
NETWORK PROPERTIES

Network	Vertex	Edge	Real Clusters
karate	34	78	2
dolphin	62	159	2
football	115	613	12
SFI	118	200	unknown
netscience	1589	2742	unknown
power grid	4941	6594	unknown

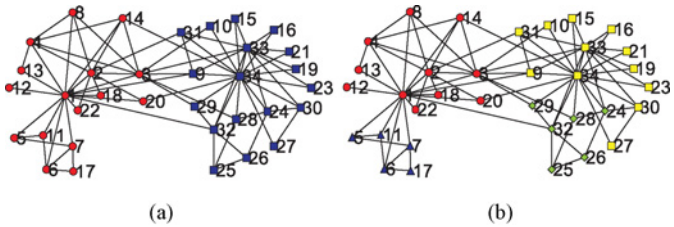


Fig. 8. Clustering results on karate club network by MODPSO. (a) Real structure detected by MODPSO. (b) Structure with highest Q value.

[67], and the power grid network [4]. The characteristics of the networks are given in Table IV.

Table V records the statistical results on the Pareto front over 30 independent trials for each algorithm on the three small networks whose ground truths are known. In addition, we perform a statistical analysis using the Welch's t-test and report the p-values in Table V.

In our hypothesis testing experiments, we perform the right-tailed t-test with the null hypothesis that the two independent samples in the vectors X and Y (the two modularity sets) come from distributions with equal means and the significance level α is set as 0.05. $\alpha \geq p$ indicates that the null hypothesis can be rejected at the 5% significance level. We can observe from the small p-values ($p \leq 0.05$) listed in Table V that our proposed algorithm outperforms almost all the comparison algorithms on the three small networks with known ground truths.

The karate network was complied by Zachary. He observed 34 members of a karate club over a period of two years. Because of fierce dispute developed between the administrator and the instructor of the club, eventually, the club is separated into two groups. From what is recorded in Table V, we can clearly notice that MODPSO can successfully detect the clustering ground truth of the network (correspond to NMI=1). It also figures out the partition with highest Q value. These two clustering situations are displayed in Fig. 8(a) and (b). It is obvious that Fig. 8(b) is a subdivision of Fig. 8(a). On this network, MOEA/D-net and MODPSO perform the best from the angle of Q, but from the perspective of statistics $p = 0.5$ indicates that the statistical difference between their performances is not significant.

The dolphin network represents a network of 62 bottlenose dolphins. A tie between two dolphins was established by their statistically significant frequent association. The network naturally is separated into two large groups, the female group and the male one. On this network, our algorithm and MOEA/D-net correctly figure out the true partition of the network, and from the perspective of modularity, our algorithm is superior

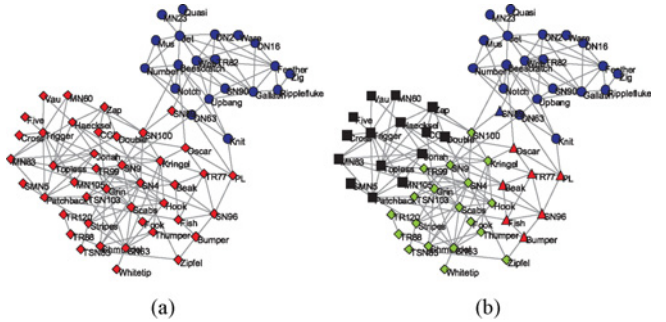


Fig. 9. Clustering results on dolphin network by MODPSO. (a) Real structure detected by MODPSO. (b) Structure with highest Q detected by MODPSO.

to MOEA/D-net but inferior to that of MOCD. The detected results by our algorithm are displayed in Fig. 9.

As is shown in Fig. 9(b) that MODPSO separates the bottom part of the real network into three smaller parts but it misplaces vertex SN89. In some literatures, SN89 can be viewed as a fuzzy node, i.e., it can be either classified to the first cluster or to the second one. GA-net has found nine clusters, but it misplaces node 8, 20, and 29. Meme-net discovers five clusters, but node 8 and 20 have been misplaced. GN reveals five communities by dividing the bottom part into four smaller parts, but it is a pity that node 40 is wrongly classified. From the p-values listed in the table, we can conclude that on this network there are no significant statistical differences between the performances of MODPSO, MOPSO-r2, and Informap.

The football network represents the American football games between Division IA colleges during the regular Fall 2000 season. Due to the self-complicated structure of the network, none of the algorithms can find the true partition. Meme-net and GA-net all wrongly separate several nodes. GN gets 12 communities, which is rather close to the real ones though node 50, 60, 64, and 98 have been forced to form a new group. In [47], the author pointed out that this is because of the nuances in the scheduling of games. Fig. 10 shows the clustering results of MODPSO on this complex network. For the partition with the highest modularity, our algorithm yields 10 clusters. By observation we find that it misplaces several vertices such as number 12, 25, 51, 36, 42, 58, 59, and 63. However, from the angles of NMI and modularity, our proposed method is still very effective and promising. From the p-values we note that MOPSO-r1 is superior to our algorithm and MOEA/D-net performs similarly good as our algorithm while our algorithm outperforms the rest methods.

Table VI records the statistical modularity values of the algorithms on the three complex networks with unknown ground truths. Statistical analysis using the Welch's t-test is also performed. Our algorithm performs remarkably well and it is very fast, especially when the size of the network is big. On the two large-scale networks, Meme-net and GN cannot give output within 40 min, which indicates they possess high computational complexity.

From these very small p-values ($p \leq 0.05$) recorded in the table, one concludes that the difference is statistically significant. From the p-values we note that our algorithm

TABLE V
EXPERIMENTAL RESULTS ON THE THREE NETWORKS WITH KNOWN GROUND TRUTHS

Network	Index	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
karate	NMI_{max}	**1**	0.8372	**1**	0.6369	**1**	**1**	0.8372	**1**	0.8630	0.6920	0.6995
	NMI_{avg}	**1**	0.8372	0.8374	0.6369	0.8598	**1**	0.8372	**1**	0.8630	0.6920	0.6995
	Q_{max}	**0.4198**	0.4151	**0.4198**	0.4059	0.4020	**0.4198**	0.4188	**0.4198**	0.2330	0.3800	0.4020
	Q_{avg}	**0.4198**	0.4112	0.4189	0.4059	0.3857	0.4160	0.4188	**0.4198**	0.2330	0.3800	0.4020
	p-value	×	0.0000	0.0042	0.0000	0.0000	0.0000	0.0000	0.5000	0.0000	0.0000	0.0000
dolphin	NMI_{max}	**1**	0.8379	**1**	0.4304	**1**	**1**	**1**	**1**	0.5540	0.5730	0.5622
	NMI_{avg}	**1**	0.8379	0.9566	0.4149	0.7853	0.9442	0.9901	**1**	0.5540	0.5730	0.5662
	Q_{max}	**0.5268**	0.5021	0.5258	0.5014	0.5155	0.5258	0.5259	0.5210	0.4060	0.4950	0.5247
	Q_{avg}	**0.5248**	0.4981	0.5237	0.4946	0.4838	0.5215	0.5210	0.5189	0.4060	0.4950	0.5247
	p-value	×	0.0000	0.0762	0.0000	0.0000	0.0000	1.0242e-4	0.0000	0.0000	0.0000	0.3491
football	NMI_{max}	**0.9289**	0.9269	0.9064	0.9194	0.8616	0.8046	0.8928	0.9269	0.9210	0.7620	0.9242
	NMI_{avg}	**0.9278**	0.9269	0.8964	0.9001	0.7739	0.7883	0.8568	0.9264	0.9210	0.7620	0.9242
	Q_{max}	**0.6046**	**0.6046**	**0.6046**	0.5940	0.5888	0.5280	0.5958	0.6044	0.5350	0.5770	0.6005
	Q_{avg}	0.6035	**0.6044**	0.6012	0.5830	0.5512	0.5173	0.5785	0.6032	0.5350	0.5770	0.6005
	p-value	×	1	0.0491	0.0000	0.0000	0.0000	0.0000	0.2188	0.0000	0.0000	0.0000

× means that there does not exist the value. The p-value is obtained over two modularity sets obtained by MODPSO and the comparison algorithm.

TABLE VI
EXPERIMENTAL RESULTS ON THE THREE NETWORKS WITH UNKNOWN GROUND TRUTHS

Network	Index	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
SFI	Q_{max}	**0.7484**	0.5877	0.5733	0.5867	0.7097	0.7430	**0.7493**	0.7312	0.7027	0.7335	0.7334
	Q_{avg}	**0.7481**	0.5823	0.5528	0.5748	0.6818	0.7323	0.7474	0.7211	0.7027	0.7335	0.7334
	p-value	×	0.0000	0.0000	0.0000	0.0000	0.0000	**0.5460**	0.0000	0.0000	0.0000	0.0000
netscience	Q_{max}	**0.9503**	0.9021	0.8381	0.8581	|	0.8916	0.8923	0.9143	|	**0.9555**	0.9252
	Q_{avg}	**0.9493**	0.9016	0.7878	0.8473	|	0.8810	0.8886	0.9060	|	**0.9555**	0.9252
	p-value	×	0.0000	0.0000	0.0000	|	0.0000	0.0000	0.0000	|	**1**	0.0000
power grid	Q_{max}	**0.8299**	0.6067	0.6362	0.6660	|	0.7035	0.7065	0.6880	|	**0.9229**	0.8140
	Q_{avg}	**0.8225**	0.6065	0.5987	0.6571	|	0.6949	0.7003	0.6815	|	**0.9229**	0.8140
	p-value	×	0.0000	0.0000	0.0000	|	0.0000	0.0000	0.0000	|	**1**	0.0217

The p-value is obtained over two modularity sets obtained by MODPSO and the comparison algorithm. | denotes that the corresponding algorithm cannot give outputs within a given time (40 min). × means that there does not exist the value.

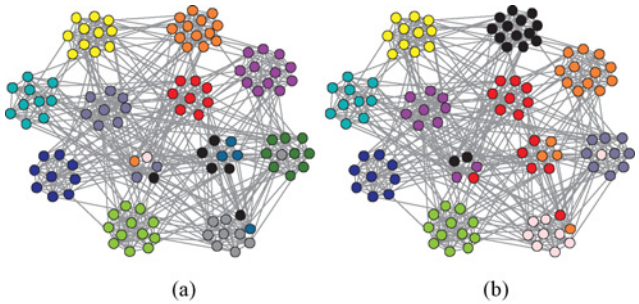


Fig. 10. Clustering results on football network by MODPSO. (a) Structure with highest NMI value. (b) Structure with highest Q value.

visibly outperforms almost all the comparison algorithms. On the SFI network, the p-value over the two modularity sets obtained by MODPSO and MOCD is 0.5460, which means that the corresponding algorithm outperforms our method. On the netscience network and the power grid network, our method outperforms all the rest algorithms except that of CNM. One explanation is that CNM starts from a set of isolated nodes, an then the links of the original graph are iteratively added such to produce the largest possible increase of the modularity. Next, we will analyze the clustering results obtained by our algorithm.

The SFI network represents 271 scientists in residence at the Santa Fe Institute, Santa Fe, NM, USA, during any part of calendar year 1999 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 component of the SFI graph consists of 118 vertices and we only do experiments on this part.

In Fig. 11, we illustrate the results from the application of our algorithm to the largest component of the SFI network and compare it with that of GN. Vertices are drawn as different shapes and colors according to the primary divisions detected.

From the figure, we clearly find that the MODPSO splits the network into eight strong communities, with the divisions running principally along disciplinary lines while GN gets seven. In Fig. 11(a), community at the top (black diamond) represents a group of scientists using agent-based models to study problems in economics and traffic flow. The second community (green triangle) represents a group of scientists working on mathematical models in ecology. The third community (circle in yellow, pink, red, and blue), which is also the biggest one, represents a group of scientists working primarily in statistical physics. MODPSO and GN subdivide this group into four small ones. The last community at the bottom of the figure is a group working primarily on the structure of RNA. MODPSO subdivides it into two small ones, just like the author mentioned in [47] that it can be divided further into smaller subcommunities, centered once again around the interests of leading members. GA-net and Meme-net detect 26

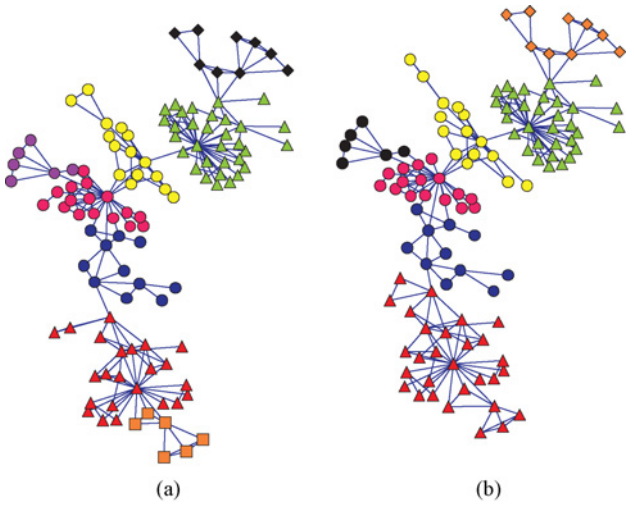


Fig. 11. Experiments on SFI network. (a) Result of MODPSO. (b) One of GN. Different colors represent different communities.

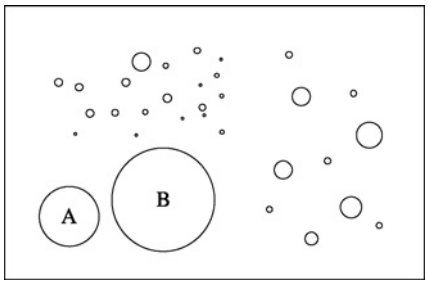


Fig. 12. Schematic example of the uncovered clusters by MODPSO on the netscience network.

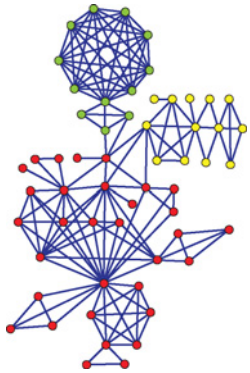


Fig. 13. Clustering result of our algorithm on the big part A of the netscience network.

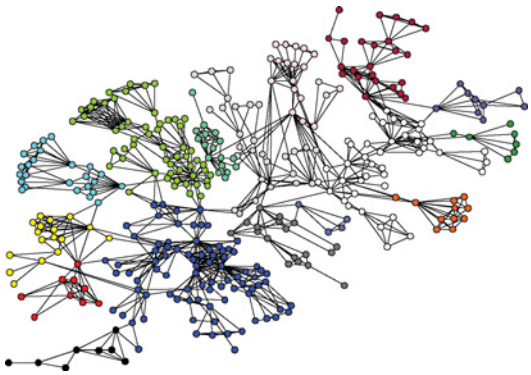


Fig. 14. Clustering result of our algorithm on the big part B of the netscience network.

and 15 communities, respectively, but through our analysis, we regard that this results remains much to be expected. Thus, we can see that MODPSO performs remarkably well.

The netscience network is a network of co-authorship of scientists working on network theory and experiment. This network is weighted, in our experiments, but we handle it as an unweighted one. When running the GN and Meme-net algorithms, they cannot give outputs within 40 min, which indicates that they have high computational complexity. The reason that leads to this phenomenon is that GN runs into calculating the edge betweenness, which is time consuming while Meme-net is involved in the greedy local search step, which is very slow. However, our algorithm can get a high modularity within a very short time.

On this network, our algorithm obtains a partition of 286 clusters with a big modularity value of 0.9503, which indicates that our algorithm has discovered communities with strong structures. The clustering results can be represented by Fig. 12.

In Fig. 12, each circle denotes one cluster; the size of the circle indicates the size of the corresponding cluster. Circles A and B are the two largest clusters uncovered by our algorithm. Figs. 13 and 14 exhibit the revealed structures of A and B.

The large-scale power grid network represents the topology of the Western States Power Grid of the United States. Each node is a power base station and the edge denotes the transforming line between two stations. On this network, all the algorithms except GN and Meme-net can yield high modularity values and our algorithm divides this big network into 255 strong small groups with a corresponding modularity value of 0.8299 though it is a little shy of that of CNM. CNM produces higher modularity value, but it still faces the modularity limitation problem while our approach just can avoid this. Besides, in [5], the author pointed out that a higher modularity does not necessarily mean a better partition of a network. What is more, we believe that to obtain just one partition for a large-scale network is not sufficient. It is better to provide more clustering results for users to choose with different preferences and our algorithm can produce a set of different solutions. In addition, our algorithm is very fast. On the author's personal computer, testing this large dataset with our proposed algorithm only requires about 5 min.

From the above experiments we see that MODPSO is remarkably promising for solving a complex network clustering problem. This is mainly due to the introduced diversity preservation mechanisms and the well-defined discrete PSO framework. Label propagation initialization can generate individuals with high clustering precision, turbulence operator helps to search optima solutions. The adopted decomposition strategy decomposes the MOP into a set of distinct scalar aggregation problems. Every particle solves the corresponding problem by applying priorities to each objective according to its weighting vector. This assists the optimization process to find potential solutions that are evenly distributed along the Pareto front and to mitigate against premature convergence.

TABLE VII
STATISTICAL RESULTS OVER 30 RUNS ON SIGNED NETWORKS

Network	Algorithm	NMI_{max}	NMI_{avg}	*clusters*	SQ_{max}	SQ_{avg}	*p*-value
illustration 1	A1	**1**	**0.9742**	3	**0.5213**	**0.5112**	×
	A2	0.7057	0.7022	8	0.3890	0.3868	0.0000
	A3	0.7495	0.7444	7	0.4305	0.4270	0.0000
	A8	0.7399	0.7399	7	0.4123	0.4115	0.0000
illustration 2	A1	**1**	**0.9959**	3	**0.5643**	**0.5634**	×
	A2	0.8213	0.8213	5	0.5214	0.5214	0.0000
	A3	0.7439	0.7126	8	0.4691	0.4532	0.0000
	A8	0.8184	0.8184	5	0.5214	0.5214	0.0000
SPP	A1	**1**	0.9949	2	**0.4547**	0.4532	×
	A2	**1**	**1**	2	**0.4547**	**0.4547**	0.8393
	A3	**1**	0.9592	2	**0.4547**	0.4424	0.0054
	A8	0.8471	0.8471	3	0.4086	0.4086	0.0000
GGS	A1	**1**	**1**	3	**0.4310**	**0.4310**	×
	A2	0.9106	0.9106	4	0.3870	0.3870	0.0000
	A3	**1**	0.9940	3	**0.4310**	0.4281	0.0777
	A8	0.9106	0.9106	4	0.3870	0.3870	0.0000

The *p*-value is obtained over two modularity sets obtained by MODPSO and the comparison algorithm. × means that there does not exist the value.

D. Experiments on Signed Networks

In this part, we will show the applications of our algorithm to two illustrative signed networks and two real-world signed networks.

1) *Illustrative signed networks:* The two illustrative networks [68] consist of 28 nodes divided into three clusters, where the links within which are positive and between which are negative.

2) *Slovene Parliamentary Party (SPP) Network:* The relation network of 10 Slovene Parliamentary parties was set up by a series of experts on Parliamentary activities in 1994 [69]. SKD, ZLSD, SDSS, LDS, ZS-ESS, ZS, DS, SLS, SPS-SNS, and SNS are the names of the ten parties respectively in short.

3) *Gahuku-Gama Subtribes (GGS) Network:* This network, established on the basis of Read's study about the cultures of highland New Guinea [70] (), describes the political alliances and oppositions among 16 Gahuku-Gama subtribes distributed in a particular area and were involved in warfare with one another in 1954.

For signed networks, we only test the performances of four algorithms, MODPSO, MOPSO-r1, MOPSO-r2, and MOEA/D-net, because the rest of the algorithms listed in this paper cannot deal with signed networks due to the limitation of objective functions. To run MOPSO-r1, MOPSO-r2, and MOEA/D-net, we change the original objective functions to those defined in our paper. Another change is the mutation possibility that is set as 0.9, because our algorithm adopts the NBM mutation operation in which every node randomly assign its label identifier to all of its neighbors. What is more, in order to promote dense connections within network clusters, we limit the NBM process to only work on positive neighbors (a node is said to have a positive neighbor if there is positive link between them); thus, the larger mutation possibility, the better.

Table VII records the statistical results over 30 trials for the algorithms on the four signed networks. Statistical analysis using the Welch's *t*-test is also performed and the *p*-values are reported.

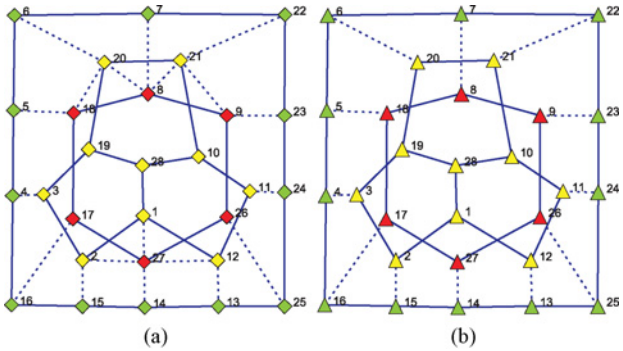


Fig. 15. Topology structures recognized by our algorithm on the illustrative networks. (a) Illustrative network 1. (b) Illustrative network 2.

From the *p*-values listed in Table VII we note that, on the two illustrative networks, our algorithm performs better than the comparison algorithms. On the SPP network, the big *p*-value indicates that MOPSO-r1 performs better than our method and, on the GGS network, our algorithm outperforms MOPSO-r1 and MOEA/D-net and the statistical differences between the performances of MODPSO, and MOPSO-r2 are not significant.

On the illustrative networks, only our algorithm can detect the ground truths of the networks. Fig. 15(a) and (b) show the clustering results of our algorithm on the two illustrative networks, where solid lines denote positive links and dashed lines denote negative links. Different colors represent different clusters. It has been found that the detected structures also have the largest *SQ* values.

On the Pareto front yielded by our algorithm on the SPP network, there are three points representing three different solutions. One solution corresponds to the true partition with highest *SQ* value shown in Fig. 16. Another solution separates the network as a single cluster and the third solution divides the network into three parts shown in Fig. 16.

From Fig. 17 we see that node SNS has been isolated as a single group. This kind of segmentation is meaningful because

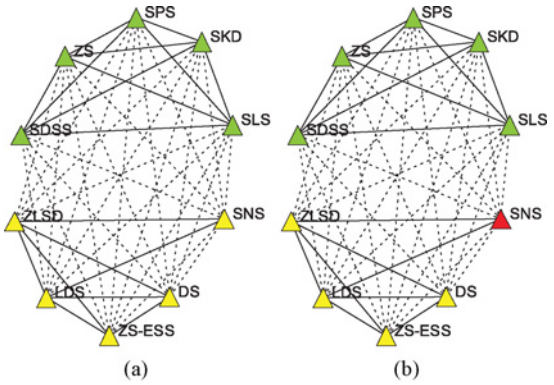


Fig. 16. Clustering results of our algorithm on the SPP network. (a) Case 1 and (b) Case 2 are the two clustering topology structures corresponding to the two solutions on the Pareto front.

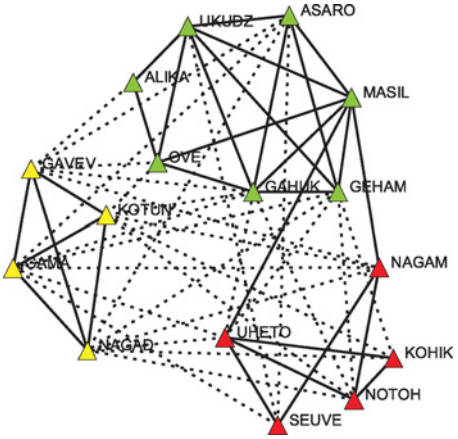


Fig. 17. Topology structures recognized by our algorithm on the GGS network.

node SNS is different from the rest nodes in its original group; it has both positive and negative links with the rest members.

The topology community structure of the GGS network recognized by our algorithm is shown in Fig. 17, where links represent political arrangements with positive (solid line) and negative (dash line) ties. Our algorithm on this network is rather stable. This is mainly due to the intrinsic strong structure of the network since there only exists positive links within group and negative links between groups.

V. Conclusion

PSO and MOPSO algorithms have been extensively studied and widely used for continuous optimization problems, but their applications in discrete space are still at low pace. This paper first proposed a discrete PSO framework in which the particle's velocity and position vectors and the arithmetical operators between them were redefined. Based on the proposed discrete PSO framework, this paper proposed a novel discrete MOPSO algorithm for solving complex network clustering problem for the first time.

The proposed MODPSO algorithm first adopted decomposition mechanism to decompose our multiobjective network clustering problem into a number of scalar problems. Then it optimized them simultaneously using a newly proposed discrete PSO framework in which the particle's representation and updating rules were redefined in discrete context. The diversity among these subproblems will naturally lead to diversity in the population. A problem-specific population initialization method based on label propagation was introduced. This method can generate diverse individuals with high clustering efficiency. A neighbor-based turbulence operator was utilized to promote diversity.

Our proposed algorithm for complex network clustering minimizes two objectives termed as kernel k-means and ratio cut. Since the two objectives leave out of consideration of the signed features of complex networks, in order to handle signed networks, we extend our objective functions into signed version. We experimentally investigated the performances of our algorithm on both signed networks and unsigned networks. Extensive experiments with ten state-of-the-art approaches, especially the network clustering algorithm with multiobjective evolutionary algorithm based on decomposition and the multiobjective particle swarm optimization method, demonstrated that our proposed algorithm is effective and promising.

Acknowledgment

The authors would like to thank Prof. Qingfu Zhang for his valuable comments and helpful suggestions, which greatly improved the paper's quality.

References

[1] M. E. J. Newman, "The structure of scientific collaboration networks," in *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, Jan. 2001.
[2] R. Albert, H. Jeong, and A. L. Barabasi, "Internet—Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.
[3] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web," *Comput. Netw.*, vol. 33, no. 1, pp. 309–320, 2000.
[4] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.
[5] S. Fortunato, "Community detection in graphs," *Phys. Rev. E*, vol. 486, no. 3, pp. 75–174, 2010.
[6] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, no. 6, p. 066133, Jun. 2004.
[7] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," in *Proc. Parallel Problem Solving Nat.*, vol. 5199. 2008, pp. 1081–1090.
[8] M. Gong, B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks," *Phys. Rev. E*, vol. 84, no. 5, p. 056101, Nov. 2011.
[9] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 418–430, Jun. 2012.
[10] C. Shi, Z. Y. Yan, Y. N. Cai, and B. Wu, "Multi-objective community detection in complex networks," *Appl. Soft Comput.*, vol. 12, no. 2, pp. 850–859, Feb. 2012.
[11] M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," *Phys. A*, vol. 391, no. 15, pp. 4050–4060, 2012.
[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[13] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. Evol. Methods Design Optimization Control Applicat. Indus. Problems*, 2002, pp. 95–100.
[14] J. D. Knowles and D. W. Corne, "Approxmating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.

[15] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 269–283, Jun. 2008.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4. Nov. 1995, pp. 1942–1948.

[17] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[18] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.

[19] J. Liang, A. Qin, P. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[20] Z. Zhu, J. Zhou, Z. Ji, and Y. Shi, "DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 643–658, Oct. 2011.

[21] Y. Gong, J. Zhang, O. Liu, R. Huang, H. Chung, and Y. Shi, "Optimizing vehicle routing problem with time windows: A discrete particle swarm optimization approach," *IEEE Trans. Syst. Man Cybern.*, vol. 42, no. 2, pp. 254–267, Mar. 2012.

[22] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.

[23] S. Z. Zhao, M. I. Willjuice, S. Baskar, and P. N. Suganthan, "Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization," *Inf. Sci.*, vol. 181, no. 16, pp. 3323–3335, Aug. 2011.

[24] L. C. Cagnina, S. C. Esquivel, and C. Coello Coello, "A fast particle swarm algorithm for solving smooth and non-smooth economic dispatch problems," *Eng. Optim.*, vol. 43, no. 5, pp. 485–505, 2011.

[25] C. Coello Coello, G. Pulido, and M. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[26] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method in multiobjective problems," in *Proc. ACM Symp. Appl. Comput.*, 2002, pp. 603–607.

[27] C. Coello Coello and M. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2. May 2002, pp. 1051–1056.

[28] X. Hu, R. Eberhart, and Y. Shi, "Particle swarm with extended memory for multiobjective optimization," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2003, pp. 193–197.

[29] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2003, pp. 26–33.

[30] L. Zhang, C. Zhou, X. Liu, Z. Ma, M. Ma, and Y. Liang, "Solving multiobjective optimization problems using particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, vol. 4. Dec. 2003, pp. 2400–2405.

[31] U. Baumgartner, C. Magele, and W. Renhart, "Pareto optimality and particle swarm optimization," *IEEE Trans. Magnetics*, vol. 40, no. 2, pp. 1172–1175, Mar. 2004.

[32] D. Srinivasan and T. Seow, "Particle swarm inspired evolutionary algorithm (PS-EA) for multi-criteria optimization problems," *Evol. Multiobjective Optimization*, vol. 3, pp. 147–165, Sep. 2005.

[33] M. Köppen and C. Veenhuis, "Multi-objective particle swarm optimization by fuzzy-Pareto-dominance meta-heuristic," *Int. J. Hybrid Intell. Syst.*, vol. 3, no. 4, pp. 179–186, 2006.

[34] L. Benameur, J. Alami, and A. El Imrani, "A new hybrid particle swarm optimization algorithm for handling multiobjective problem using fuzzy clustering technique," in *Proc. Int. Conf. Comput. Intell. Model. Simul. CSSim'09*, Sep. 2009, pp. 48–53.

[35] Z. Li, R. Ngambusabongsopa, E. Mohammed, and N. Eustache, "A novel diversity guided particle swarm multi-objective optimization algorithm," *Int. J. Digital Content Technol. Applicat.*, vol. 5, no. 1, pp. 269–278, 2011.

[36] M. Daneshyari and G. Yen, "Cultural-based multiobjective particle swarm optimization," *IEEE Trans. Syst. Man Cybern.*, vol. 41, no. 2, pp. 553–567, Apr. 2011.

[37] N. Al Moubayed, A. Petrovski, and J. McCall, "D^2MOPSO: Multi-objective particle swarm optimizer based on decomposition and dominance," *Proc. 12th Eur. Conf. Evol. Comput. Combinatorial Optimization*, vol. 7245, pp. 75–86, Apr. 2012.

[38] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 5. Oct. 1997, pp. 4104–4108.

[39] A. Salman, I. Ahmad, and S. Al-Madani, "Particle swarm optimization for task assignment problem," *Microprocessors Microsyst.*, vol. 26, no. 8, pp. 363–371, 2002.

[40] D. Y. Sha and C. Y. Hsu, "A hybrid particle swarm optimization for job shop scheduling problem," *Comput. Ind. Eng.*, vol. 51, no. 4, pp. 791–808, 2006.

[41] M. Mitrović and B. Tadić, "Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities," *Phys. Rev. E*, vol. 80, no. 2, p. 026123, Aug. 2009.

[42] W. Pang, K. P. Wang, C. G. Zhou, and L. J. Dong, "Fuzzy discrete particle swarm optimization for solving traveling salesman problem," in *Proc. 4th Int. Conf. Comput. Inform. Technol.*, Sep. 2004, pp. 796–800.

[43] C. J. Liao, C. T. Tseng, and P. Luarn, "A discrete version of particle swarm optimization for flowshop scheduling problems," *Comput. Oper. Res.*, vol. 34, no. 10, pp. 3099–3111, 2007.

[44] W. Chen, J. Zhang, H. Chung, W. Zhong, W. Wu, and Y. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278–300, Apr. 2010.

[45] P. Doreian and A. Mrvar, "A partitioning approach to structural balance," *Social Netw.*, vol. 18, no. 2, pp. 149–168, 1996.

[46] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," in *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 9, pp. 2658–2663, Mar. 2004.

[47] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," in *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[48] S. Gómez, P. Jensen, and A. Arenas, "Analysis of community structure in networks of correlated data," *Phys. Rev. E*, vol. 80, p. 016114, Jul. 2009.

[49] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," in *Proc. Natl. Acad. Sci. USA*, vol. 104, no. 1, pp. 36–41, 2007.

[50] Z. Li, S. Zhang, R. Wang, X. Zhang, and L. Chen, "Quantitative function for community detection," *Phys. Rev. E*, vol. 77, no. 3, p. 036109, 2008.

[51] A. Lancichinetti, S. Fortunato, and K. Kertesz, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, p. 033015, 2009.

[52] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E*, vol. 74, no. 1, p. 016110, 2006.

[53] A. Arenas, A. Fernández, and S. Gómez, "Analysis of the structure of complex networks at different resolution levels," *New J. Phys.*, vol. 10, no. 5, p. 053039, 2008.

[54] P. Pons and M. Latapy, "Post-processing hierarchical community structures: Quality improvements and multi-scale view," *Theoretical Comput. Sci.*, vol. 412, no. 8, pp. 892–900, 2011.

[55] L. Angelini, S. Boccaletti, D. Marinazzo, M. Pellicoro, and S. Stramaglia, "Identification of network modules by optimization of ratio association," *Chaos*, vol. 17, no. 2, p. 023114, 2007.

[56] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[57] M. Gong, Q. Cai, Y. Li, and J. Ma, "An improved memetic algorithm for community detection in complex networks," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 1–8.

[58] M. Villalobos-Arias, G. Pulido, and C. Coello Coello, "A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimizer," in *Proc. IEEE Swarm Intell. Symp.*, Jun. 2005, pp. 22–29.

[59] G. Palermo, C. Silvano, and V. Zaccaria, "Discrete particle swarm optimization for multi-objective design space exploration," in *Proc. 11th EUROMICRO Conf. Digital Syst. Design Architectures Methods Tools*, 2008, pp. 641–644.

[60] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, no. 6, p. 066111, 2004.

[61] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," in *Proc. Natl. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, 2008.

[62] F. Wu and B. A. Huberman, "Finding communities in linear time: A physics approach," *Eur. Phys. J. B*, vol. 38, no. 2, pp. 331–338, 2004.

[63] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Phys.*, vol. 2005, no. 09, p. P09008, 2005.

[64] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E*, vol. 78, p. 046110, Oct. 2008.
[65] W. W. Zachary, "An information flow model for confict and fission in small groups," *J. Anthropol. Res.*, pp. 452–473, 1977.
[66] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405, 2003.
[67] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, p. 036104, Sep. 2006.
[68] B. Yang, W. K. Cheung, and J. M. Liu, "Community mining from signed social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 10, pp. 1333–1348, Oct. 2007.
[69] S. Kropivnik and A. Mrvar, "An analysis of the slovene parliamentary parties network," in *Developments in Statistics and Methodology*, A. Ferligoj, A. Kramberger, Eds. Metodološki zvezki, 12, Ljubljana: FDV, 1996, pp. 209–216.
[70] K. E. Read, "Cultures of the central highlands, new guinea," *Southwestern J. Anthropology*, vol. 10, no. 1, pp. 1–43, 1954.

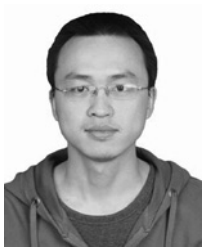
Maoguo Gong (M'07) received the B.S. degree in electronic engineering (first class honors) and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively.

Since 2006, he has been a Teacher with Xidian University. He is currently a Full Professor with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University. His current research interests include computational intelligence with applications.

Dr. Gong is a Member of the IEEE Computational Intelligence Society, an Executive Committee Member of the Natural Computation Society of Chinese Association for Artificial Intelligence, and a Senior Member of the Chinese Computer Federation. He was the recipient of the New Century Excellent Talent in University Award of the Ministry of Education of China, the Eighth Young Scientist Award of Shaanxi, the New Scientific and Technological Star of Shaanxi Province, and the Science and Technology Award of Shaanxi Province (first level, 2008 and 2010).

Qing Cai received the B.S. degree in electronic information engineering from Wuhan Textile University, Wuhan, China, in 2010. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems at the School of Electronic Engineering, Xidian University, Xi'an, China.

His research interests include the area of computational intelligence and complex network analysis.



Xiaowei Chen received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2011. He is currently pursuing the M.S. degree at the School of Electronic Engineering, Xidian University.

His research interests include community detection in networks, multiobjectiove optimization, and memetic algorithms.

Lijia Ma received the B.S. degree in communication engineering from Hunan Normal University, Hunan, China, in 2010. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems at the School of Electronic Engineering, Xidian University, Xi'an, China.

His research interests include evolutionary multiobjective optimization, data mining, and complex network analysis.