Discrete particle swarm optimization for identifying community structures in signed social networks

Article *in* Neural Networks · October 2014

DOI: 10.1016/j.neunet.2014.04.006

Qing Cai
Xidian University

Ma Lijia
Nanyang Technological University

Neural Networks 58 (2014) 4–13

Contents lists available at ScienceDirect

Neural Networks

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

2014 Special Issue

Discrete particle swarm optimization for identifying community structures in signed social networks

Qing Cai, Maoguo Gong *, Bo Shen, Lijia Ma, Licheng Jiao

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi Province 710071, China

ARTICLE INFO

Article history:
Available online 13 May 2014

Keywords:
Signed social network
Community detection
Particle swarm optimization
Evolutionary algorithm

ABSTRACT

Modern science of networks has facilitated us with enormous convenience to the understanding of complex systems. Community structure is believed to be one of the notable features of complex networks representing real complicated systems. Very often, uncovering community structures in networks can be regarded as an optimization problem, thus, many evolutionary algorithms based approaches have been put forward. Particle swarm optimization (PSO) is an artificial intelligent algorithm originated from social behavior such as birds flocking and fish schooling. PSO has been proved to be an effective optimization technique. However, PSO was originally designed for continuous optimization which confounds its applications to discrete contexts. In this paper, a novel discrete PSO algorithm is suggested for identifying community structures in signed networks. In the suggested method, particles' status has been redesigned in discrete form so as to make PSO proper for discrete scenarios, and particles' updating rules have been reformulated by making use of the topology of the signed network. Extensive experiments compared with three state-of-the-art approaches on both synthetic and real-world signed networks demonstrate that the proposed method is effective and promising.

1. Introduction

The modern science of social networks is an active domain within the new interdisciplinary science of complex systems. In reality, many intricate systems can be represented as social networks, such as the complex collaboration networks (Newman, 2001), the world-wide-web (Albert, Jeong, & Barabasi, 1999; Broder et al., 2000), etc. Social networks are characterized by big data volume, dynamics and heterogeneous, especially for the Internet based social networks. Most of the social network data are natural language based. Data mining from social networks based on natural language is challenging (Cook & Holder, 2006; Kleinberg, 2007). Recent years, opinion mining and sentiment analysis become two of the most studied tasks of natural language processing, and have gathered momentum on both theoretical and empirical studies (Cambria, Schuller, Xia, & Havasi, 2013; Gangemi, Presutti, & Reforgiato Fortunato(2010), 2014; Poria et al., 2013).

One effective way to discover knowledge from a social network is to first model the network as a graph that is composed of a set of vertices and edges, where nodes represent the objects and links represent the interactions amongst them, and then apply certain techniques to analyze the properties of the graph based network. Network has many salient properties and amongst which the community structure is believed to be an eminent feature of networks (Girvan & Newman, 2002). In the academic domain, communities, also called clusters or modules, are groups of vertices which probably share common properties and/or play similar roles within the graph. Community detection is such a tool that helps to identify community structures in networks. Network community detection is of great significance. For example, mining cybercriminal networks from online social networks can facilitate cybercrime forensics so as to reduce the financial loss (Lau, Xia, & Ye, 2014). A recent survey on network community structure mining can be found in Fortunato (2010).

Mostly, the detection of community structures in networks can be considered either as a clustering problem or an optimization problem (Newman, 2004), thus, the choice of an appropriate evaluation function affects the ultima detection performance. For this purpose, Girvan and Newman (2002) had put forward the concept of modularity as a criterion to stop the division of a network into sub-networks in their divisive hierarchical clustering algorithm based on the iterative removal of edges with high betweenness. Based on modularity, other algorithms also appear in

* Corresponding author. Tel.: +86 029 88202661; fax: +86 029 88201023.
E-mail address: gong@ieee.org (M. Gong).

http://dx.doi.org/10.1016/j.neunet.2014.04.006
0893-6080/

the literature. Guimerò, Sales-Pardo, and Anaral (2004) employed simulated annealing for modularity optimization. Extremal optimization was used for modularity optimization by Duch and Arenas (2005). Spectral optimization technique has also been utilized to optimize modularity by replacing the Laplacian matrix with the modularity matrix (Newman, 2006). Besides modularity based methods, spectral clustering methods (Mitrovic & Tadic, 2009), dynamic approaches such as spin models (Reichardt & Bornholdt, 2004), random walk (Hughes, 1995) and synchronization (Boccaletti, Ivanchenko, Latora, Pluchino, & Rapisarda, 2007), methods based on statistical inference such as block modeling (Reichardt & White, 2007) and information theory (Ziv, Middendorf, & Wiggins, 2005), have all found their niche in this area.

Evolutionary algorithms (EAs), because of their inherent global searching abilities, hold an important position in the computational intelligence domain. EAs have been proved to be effective tools for solving optimization problems. Recently, some scholars have successfully applied either single or multiobjective evolutionary algorithms to discover community structures in networks. Pizzuti (2008, 2012) proposed a single objective genetic algorithm and a multi-objective genetic algorithm, respectively. In Gong, Fu, Jiao, and Du (2011) and Ma, Gong, Liu, Cai, and Jiao (2014) we had proposed a Memetic algorithm for community detection, and in Gong, Cai, Chen, and Ma (2014) we had suggested a multiobjective particle swarm optimization based approach.

In the field of social science, social networks with both positive and negative links are called signed networks (Doreian & Mrvar, 1996). In a signed network, the positive links denote "positive relationships" such as "friendship, common interests" and the negative links may denote "negative relationships" such as "hostility, different interests". To probe community structures in signed networks will shed light on how real society operates. Yang et al. have proposed a Markov random walk based algorithm called FEC to mine community structures in signed networks in Yang, Cheung, and Liu (2007). Doreian has designed an evaluation index to measure the quality of the partition of a signed network in Doreian (2008) and Traag and Bruggeman have applied the Potts Model (Wu, 1982) to solve the signed network community detection problem in Traag and Bruggeman (2009). Modularity is a very popular metric for community detection. Gómez, Jensen, and Arenas (2009) presented a reformulation of modularity that allows the analysis of signed networks. However, to maximize modularity is proved to be NP-hard (Brandes et al., 2006).

Although, in recent years many EAs based approaches have been developed to disclose communities in social networks, seldom of them have paid attention to signed networks. In this paper, we have newly suggested a particle swarm optimization (PSO) based algorithm to discover communities in signed social networks. PSO (Kennedy & Eberhart, 1995), originated from social animals' behavior, is well known by its fast convergence and has been proved to be one of the most popular optimization techniques. Due to its effectiveness and extremely easy implementation, PSO is gathering attention and it has found nationwide applications in diverse domains (Kiranyaz, Ince, Yildirim, & Gabbouj, 2009; Sharafi & ELMekkawy, 2014; Xu, Venayagamoorthy, & Wunsch, 2007). PSO works with a swarm of particles. Each particle adjusts its velocity by learning from its neighbors. This process is carried on simultaneously. Each particle can be seen as independent agents evolving in parallel, with some synchronizations. Thus, PSO can be regarded as an implicit parallel and distributed computational optimization algorithm, which makes it capable to handle large scale global optimization problems.

PSO is originally designed for continuous optimization which confounds its applications. In this paper, we redefine the particles' velocity and position and the main arithmetic operators between them in discrete form, consequently, a discrete PSO algorithm designed for identifying community structures in signed networks is proposed for the first time. Our method makes full use of networks' prior knowledge such as node degree information and linkage correlations. In order to speed up the algorithm convergence, a novel particle swarm initialization mechanism proposed in our previous work in Gong, Cai, Li, and Ma (2012) is adopted in this paper. Extensive experiments on both synthetic and real-world signed networks prove that the proposed algorithm is more efficient and much faster than several state-of-the-art approaches.

The rest of this paper is organized as follows. Section 2 gives the related background. In Section 3, the proposed method is presented in detail. Section 4 shows the experimental studies of the proposed method, and the conclusions are finally summarized in Section 5.

2. Related background

2.1. Community definition

The task for community detection is to separate the whole network into small parts which are called communities. In the literature, communities are regarded as subgraphs which have dense intra-links and sparse inter-links. Radicchi, Castellano, Cecconi, Loreto, and Parisi (2004) gave a community definition based on the node degree, but the community of a signed network is defined not only by the density of links but also by the signs of links. In Gong et al. (2014) we have suggested a signed community definition.

Given a signed network modeled as $G = (V, PL, NL)$, where V is the set of nodes and PL and NL are the set of positive and negative links, respectively. Let A be the adjacency matrix of G and l_{ij} be the link between node i and j. Then the element of A is defined as:

$$\begin{cases} A_{ij} = 1 & \text{if } l_{ij} \in PL \\ A_{ij} = -1 & \text{if } l_{ij} \in NL \\ A_{ij} = 0 & \text{if } \nexists l_{ij}. \end{cases} \tag{1}$$

Given that $S \subset G$ is a subgraph where node i belongs to. Let $(d_i^+)^{\text{in}} = \sum_{j \in S, l_{ij} \in PL} A_{ij}$ and $(d_i^-)^{\text{in}} = \sum_{j \in S, l_{ij} \in NL} |A_{ij}|$ be the positive and negative internal degrees of node i, respectively. Then S is a signed community in a strong sense if

$$\forall i \in S, \quad (d_i^+)^{\text{in}} > (d_i^-)^{\text{in}}. \tag{2}$$

Let $(d_i^-)^{out} = \sum_{j \notin S, l_{ij} \in NL} |A_{ij}|$ and $(d_i^+)^{out} = \sum_{j \notin S, l_{ij} \in PL} A_{ij}$ be the negative and positive external degrees of node i, respectively. Then S is a signed community in a weak sense if

$$\begin{cases} \sum_{i \in S} (d_i^+)^{\text{in}} > \sum_{i \in S} (d_i^+)^{out} \\ \sum_{i \in S} (d_i^-)^{out} > \sum_{i \in S} (d_i^-)^{\text{in}}. \end{cases} \tag{3}$$

Thus, in a strong sense, a node has more positive links than negative links within the community; in a weak sense, the positive links within a community are dense while the negative links between different communities are also dense.

The above definitions only give the conditions that a signed community should satisfy. In order to give a quantitative standard, Gómez et al. (2009) presented a reformulation of modularity that allows the analysis of signed networks. The signed modularity (SQ) is formulized as:

$$SQ = \frac{1}{2w^+ + 2w^-} \sum_{i,j} \left(w_{ij} - \left(\frac{w_i^+ w_j^+}{2w^+} - \frac{w_i^- w_j^-}{2w^-} \right) \right) \delta(i,j) \tag{4}$$

where w_{ij} is the weight of the signed adjacency matrix, $w_i^+ (w_i^-)$ denotes the sum of all positive (negative) weights of node i. If node i and j are in the same group, $\delta(i,j) = 1$, otherwise, 0. Normally by assumption we take it that the larger the value of SQ, the better the community structure is.

2.2. Particle swarm optimization

PSO, originated from the social behavior such as fish schooling and bird flocking, was first introduced in 1995 by Kennedy and Eberhart (1995). It is a population-based stochastic optimization technique and has now become one of the most popular optimization techniques (Clerc & Kennedy, 2002; Liang, Qin, Suganthan, & Baskar, 2006; Shi & Eberhart, 1998, 2001; van den Bergh & Engelbrecht, 2004).

Each individual, which is typically called a "particle" in PSO, has a position and a velocity vector. The position vector usually simulates a candidate solution to the optimization problem. In order for PSO to search for the optimal solution, a particle updates its flying trajectory with some simple rules. Provided that the particle swarm size is m and the particle dimension is n. Let $V_i = \{v_1, v_2, \dots, v_n\}$ and $X_i = \{x_1, x_2, \dots, x_n\}$ be the ith ($i = 1, 2, \dots, m$) particle' velocity and position vectors, respectively. Then in the basic form, a particle adjusts its status according to the following rules:

$$V_i \leftarrow V_i + c_1 r_1 (Pbest_i - X_i) + c_2 r_2 (Gbest - X_i) \tag{5}$$

$$X_i \leftarrow X_i + V_i \tag{6}$$

where $Pbest_i = \{pbest_1, pbest_2, \dots, pbest_n\}$ and $Gbest = \{gbest_1, gbest_2, \dots, gbest_n\}$ are the ith particle's personal best position and the best position of the swarm, respectively. c_1 and c_2 are acceleration coefficients termed as cognitive and social components. r_1 and r_2 are random numbers between 0 and 1.

Shi and Eberhart (1998) first introduced an inertia weight to the velocity updating rule which reads:

$$V_i \leftarrow \omega V_i + c_1 r_1 (Pbest_i - X_i) + c_2 r_2 (Gbest - X_i). \tag{7}$$

The authors argue that a relatively large inertia weight ω is better for global search, while a small ω enhances the ability of local search. In order to balance the global and local search ability, in Shi and Eberhart (1999) they suggested a linearly decreasing inertia weight factor which can be expressed as:

$$\omega(k) = (\omega_{\max} - \omega_{\min}) \frac{k_{\max} - k}{k_{\max}} + \omega_{\min} \tag{8}$$

where $k_{\max}$ is the number of the maximum iterations, k is the current iteration; $\omega_{\max}$ and $\omega_{\min}$ are the maximum and the minimum inertia weight value respectively. They also have suggested fuzzy methods to adjust ω nonlinearly in Shi and Eberhart (2001). Some other inertia weight adjustion strategies can be found in Ting, Shi, Cheng, and Lee (2012) and Zhou and Shi (2011).

Because of the high efficiency of PSO, much effort has been devoted to extending continuous PSO to discrete ones. The first trial was the binary PSO (BPSO) algorithm proposed by Kennedy and Eberhart (1997) based on the binary coding scheme. Later, the algorithm was improved by the angle modulated PSO (Pampara, Franken, & Engelbrecht, 2005) and the discrete multiphase PSO (Al-kazemi & Mohan, 2006). One direct way to turn continuous PSO into discrete one is to map the consecutive space into discrete form. Based on this idea, many discrete PSO algorithms characterized by space transformation technique have been put forward (Salman, Ahmad, & Al-Madani, 2002; Sha & Hsu, 2006). Other avenues based on redefining the position and velocity of a particle, such as fuzzy matrix based (Liao, Tseng, & Luarn, 2007), swap-operator-based (Mitrovic & Tadic, 2009), crisp set based (Chen et al., 2010), etc., have also been advanced in the literature. Although various discrete PSO algorithms have been proposed, their performances are generally not satisfactory when applied to the community detection problem.

3. The proposed algorithm for community detection

In this paper, by taking the advantage of network prior knowledge, we have suggested a novel discrete PSO algorithm for signed network community detection. The proposed algorithm optimizes the signed modularity index. This section will illustrate the advanced algorithm in detail.

3.1. General framework

The whole framework of the proposed discrete particle swarm optimization method for community detection (DPSO) is given in Algorithm 1.

Algorithm 1 Framework of the proposed DPSO.

Parameters: particle swarm size m, number of iterations $gmax$, inertia weight ω, learning factors c_1 and c_2;
Input: network adjacency matrix A;
Output: best fitness, partition of the signed network;

1. **Begin**
2. Population initialization: initialize position vectors X, i.e., $pop.X \leftarrow label\_propagation$; initialize velocity vectors V, i.e., $pop.V = 0$; calculate particles' fitness $pop.fitness$; update current best position vectors $pbest$ and global best position vectors $gbest$;
3. **For** $iter = 0$; $iter < gmax$; $iter++$
   (a) **For** each particle $i \in m$ **do**
      i. update particle i's status according to Eqs. (9) and (10);
      ii. rearrange particle i's position vector; // see subsection 3.5 for more information
      iii. evaluate particle i's fitness;
      iv. update particle i's *pbest* vector;
   (b) **End**
   (c) update the *gbest* vector;
4. **End**
5. **End**

3.2. Particle swarm initialization

As shown in Algorithm 1, the population initialization step needs to initialize particles' position and velocity vectors and update the *pbest* and *gbest* vectors.

To initialize the position vectors, a novel mechanism based on label propagation is introduced. This method can generate individuals with higher clustering efficiency. For more information, please refer to our previous work in Gong et al. (2012). The *pbest* vectors are initialized the same as the position vectors and the *gbest* vector is set as the best position vector in the current population.

3.3. Definition of discrete particle status

In this paper, we redefine the term "position" and "velocity" under discrete context. The definitions are as follows:

Definition of discrete position: the position vector represents a partition of a signed network. We define a position permutation in the form of $X = \{x_1, x_2, \dots, x_n\}$ where $x_i \in [1, n]$ is an integer and n is the number of the nodes of a network. If $x_i = x_j$, then we take it that node j and i are in the same community. A graphical illustration of the particle representation schema is shown in Fig. 1.

The above position definition does not need to know about the topology structures of the networks, which makes it transplantable to other kinds of networks. The representation schema is easy to decode which will lower down the computational complexity and

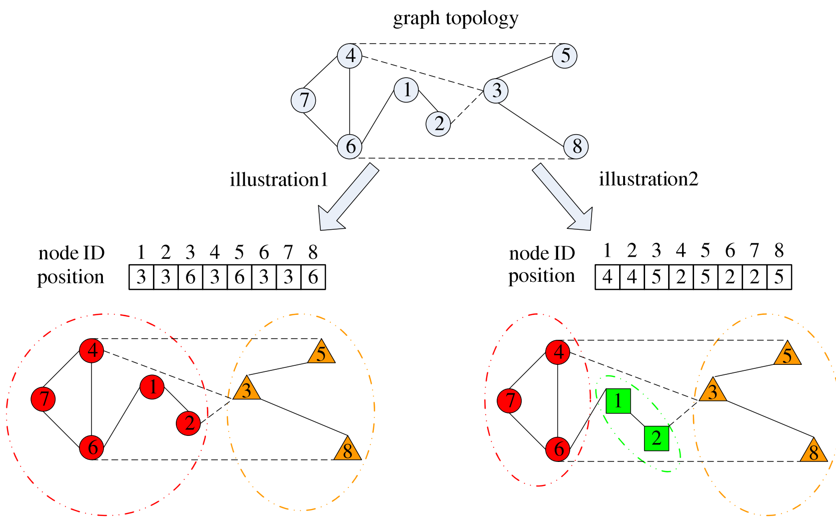


Fig. 1. A schematic illustration of the defined particle representation schema.

it can automatically determine the communities of the network, what is more, it is easy to implement label propagation operation to generate initial solutions with high clustering precision.

Definition of discrete velocity: we define the velocity permutation of a particle as $V = \{v_1, v_2, \ldots, v_n\}$ where $v_i \in \{0, 1\}$. If $v_i = 1$, then the corresponding element x_i in the position vector X will be changed according to Eq. (12), otherwise, x_i keeps unchanged.

In the canonical version of PSO, if the speed of a particle is very large, it will fly apart. To prevent a particle from flying out of the boundaries, it needs a threshold V_{max} to pull back the particle. The threshold is hard to tune. However, from the above definition of velocity we can notice that this threshold parameter is no longer needed.

3.4. Definition of particle status update rules

In PSO, a particle adjusts its velocity by learning from its neighbors. The new velocity helps the particle to fly to promising region. The particle status update rules defined in the canonical version of PSO no longer fit the discrete context, therefore, we redefine the update principles based on the newly defined particle status. The redefined particle status update rules are as follows:

$$V_i \leftarrow \Gamma(\omega V_i + c_1 r_1 (Pbest_i \oplus X_i) + c_2 r_2 (Gbest \oplus X_i)) \tag{9}$$

$$X_i \leftarrow X_i \Theta V_i. \tag{10}$$

In the above equations, the inertia weight ω is generated according to Eq. (8). In our algorithm, ω_{max} and ω_{min} are set as 0.9 and 0.4, respectively, and the learning factors c_1 and c_2 are set as 1.494.

In Eq. (9), the symbol "$\oplus$" is the XOR operator. From the perspective of swarm intelligence, a particle will adjust its velocity by learning from its neighbors. The learning process is actually a comparison between the positions. From the perspective of graph theory, the defined "$\oplus$" operation actually reflects the difference between two network structures. Therefore, we define the operation between two positions as the XOR operator. The function $\Gamma(\cdot)$ is defined as:

$$\Gamma(x) = \begin{cases} 1 & if\ x \geq 1 \\ 0 & if\ x < 1. \end{cases} \tag{11}$$

The velocity represents the tendency for the position to change. Since our defined discrete particle position works with a binary coded velocity vector, the defined function $\Gamma(\cdot)$ can transform the real coded velocity into binary form. Moreover, the definition of $\Gamma(\cdot)$ is straight forward and easy to carry out.

The operator "Θ" in Eq. (10) is the key component of the newly defined particle status update rules. The operator "Θ" works between a position and a velocity vector. When designing a good operator "Θ", we should make use of the network prior knowledge so as to drive a particle to promising region.

In our DPSO algorithm, the operator "Θ", to some extent, can be regarded as a neighbor based operation since the update of a position is based on the neighbors of a node. Given a position $X = \{x_1, x_2, \ldots, x_n\}$ and a velocity $V = \{v_1, v_2, \ldots, v_n\}$. Operator "$\Theta$" is defined as follows:

$$\begin{cases} X \Theta V = X' = \{x'_1, x'_2, \ldots, x'_n\} \\ x'_i = x_i if\ v_i = 0 \\ x'_i = \arg\max_r \sum_{j \in PN} \delta(x_j, r) if\ v_i = 1 \end{cases} \tag{12}$$

where $PN = \{pn_1, pn_2, \ldots, pn_k\}$ is the positive neighbor (have positive links with node i) set of vertex i. The symbol r is an integer. The function $\arg\max_r f(r)$ returns the value r that can lead to biggest value of $f(r)$. The essence of the symbol "Θ" is that we use the information of the dominated node in the positive neighbors to update particle's position. If there are multiple r which satisfies the maximum, then we randomly choose one from them. A graphical illustration of the "Θ" operation is shown in Fig. 2.

In Fig. 2 the fourth element of the velocity vector is 1. According to our defined update rules, the fourth element in the position vector will be changed. Since node 4 has three positive neighbors, i.e., node 1, 3 and 7, therefore, according to Eq. (12), the fourth element in the previous position vector will be changed to be 5.

3.5. Particle position rearrangement

In our proposed algorithm, a particle' position has been defined as an integer permutation. Each position permutation corresponds to a community structure of a network. However, there exists the situations that different position permutations are structurally equivalent, i.e., they correspond to the same community structure. For example, given two position vectors $P_1 = \{3\,1\,3\,1\,3\,1\}$ and $P_2 = \{6\,4\,6\,4\,6\,4\}$, then according to our defined particle representation schema, P_1 and P_2 are structurally equivalent because they both correspond to the same network partition that nodes 1, 3 and 5 belong to one community and nodes 2, 4 and 6 belong to another community.

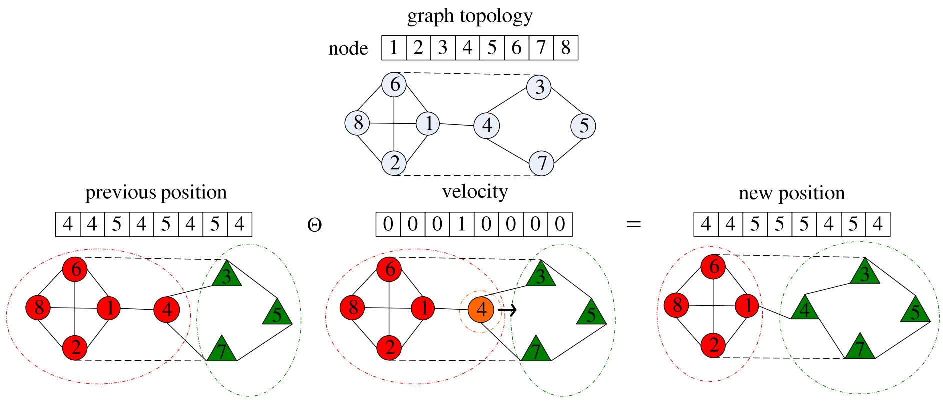


Fig. 2. A generic illustration of the defined "Θ" operation.

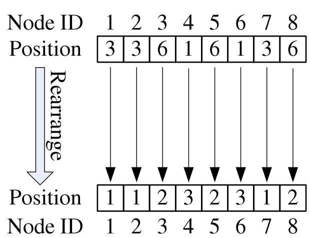


Fig. 3. A graphical illustration of the position rearrangement operation.

To avoid unnecessary computing, we design a particle position rearrangement operation in our algorithm. A graphical illustration of the rearrangement operation is shown in Fig. 3.

In Fig. 3, when we decode the original position vector we will divide the eight nodes into three communities, i.e., $c_1 = \{1, 2, 7\}$, $c_2 = \{3, 5, 8\}$ and $c_3 = \{4, 6\}$. Then we rearrange the position in the following way: the 1st, 2nd and 7th elements in the vector have been changed to be 1; the 3rd, 5th and 8th elements have been changed to be 2; the 4th and the 6th elements have been changed to be 3.

From what is illustrated above we can see that the framework of the proposed DPSO algorithm is concise and the main components of the algorithm are easy to carry out. All these merits make the advanced algorithm possible to handle large scale social networks.

4. Experimental studies

In this section, we apply our algorithm to both synthetic signed networks and real-world signed networks. And we also compare our algorithm with several state-of-the-art community detection algorithms. Our algorithm is coded in C++, the experiments have been performed on an Inter(R) Celeron(R)M CPU 550 machine, 3.2 GHz, 4 GB memory. The operating system is MS Windows XP and the compiler is VC++ 6.0.

4.1. Evaluation metric

If the ground truth of a network is known, in addition to the signed modularity index, we also adopt the widely used *Normalized Mutual Information* (*NMI*) metric (Wu & Huberman, 2004) to estimate the similarity between the true community structures and the detected ones.

Given that A and B are two partitions of a network, then $NMI(A, B)$ is written as:

$$NMI = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{C_A} C_{i.}\log(C_{i.}/N) + \sum_{j=1}^{C_B} C_{.j}\log(C_{.j}/N)} \quad (13)$$

where N is the number of nodes of the network, C is a confusion matrix. C_{ij} equals to the number of nodes shared in common by community i in partition A and by community j in partition B. C_A (or C_B) is the number of clusters in partition A (or B), $C_{i.}$ (or $C_{.j}$) is the sum of elements of C in row i (or column j). NMI is a similarity measure proved to be reliable by Danon, Diaz-Guilera, Duch, and Arenas (2005).

Table 1
Parameters of the algorithms. "*pop*" represents the population size, "*gmax*" denotes the maximum iterations of the algorithm, "*pc*" and "*pm*" are the crossover and mutation possibility, respectively, "*ns*" is the neighborhood size.

Algorithm	*pop*	*gmax*	*pc*	*pm*	*ns*	Reference
DPSO	100	100	–	–	–	[ours]
GA-net	100	100	0.9	0.1	–	Pizzuti (2008)
MOCD	100	100	0.9	0.1	–	Shi et al. (2012)
MODPSO	100	100	0.9	0.1	40	Gong et al. (2014)

Table 2
Experimental results on the synthetic signed network 1.

Algorithm	SQ_{max}	SQ_{avg}	NMI_{max}	NMI_{avg}	Cluster
DPSO	0.5213	0.5213	1	1	3
GA-net	0.5213	0.5213	1	1	3
MOCD	0.3890	0.3890	0.7057	0.7057	8
MODPSO	0.5213	0.5112	1	0.9742	3

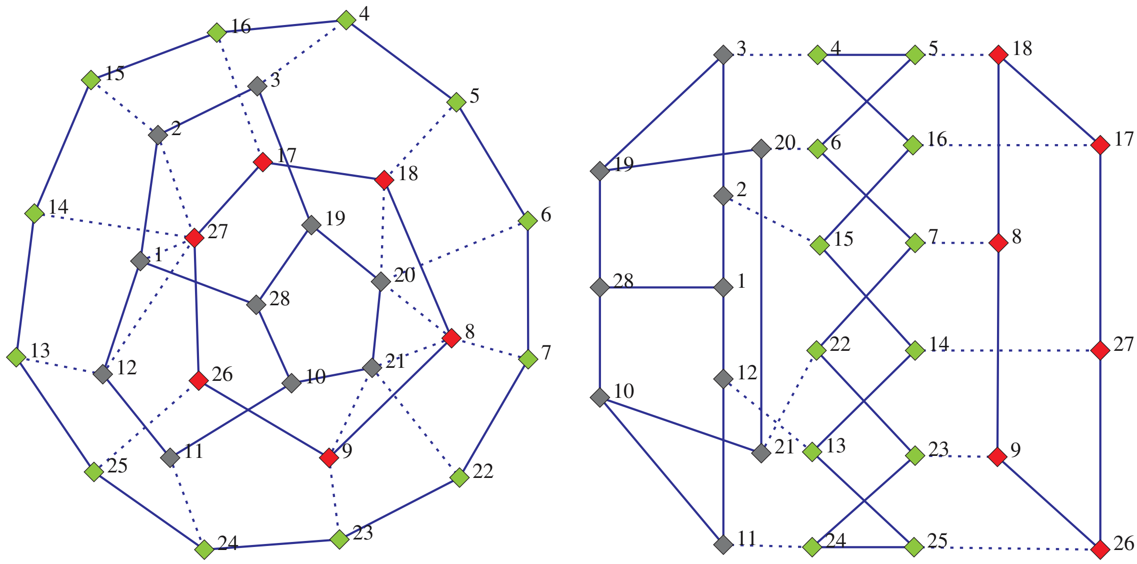
4.2. Comparison algorithms

In this paper several EA-based algorithms named as GA-net (Pizzuti, 2008), MOCD (Shi, Yan, Cai, & Wu, 2012) and MODPSO (Gong et al., 2014) are chosen to compare with the proposed algorithm. For fair comparison, the objective function used in GA-net has been changed to be the same as that used in DPSO. The experimental parameters of the four algorithms are listed in Table 1.

4.3. Results on synthetic signed networks

We first do some experiments on two synthetic signed networks introduced in Yang et al. (2007). The two synthetic signed networks consist of 28 nodes divided into three clusters. We test each of the algorithms 30 times and the maximal and averaged values of SQ and NMI over 30 runs are recorded in Tables 2 and 3. We also record the numbers of communities that correspond to the maximum SQ.

It can be seen from Tables 2 and 3 that the proposed algorithm DPSO can detect community structures of the two synthetic signed networks with the NMI metric equals 1, i.e., it can discover the



(a) Synthetic network 1.

(b) Synthetic network 2.

Fig. 4. The community structures of the synthetic signed networks detected by DPSO.

Table 3
Experimental results on the synthetic signed network 2.

Algorithm	SQ_{max}	SQ_{avg}	NMI_{max}	NMI_{avg}	Cluster
DPSO	0.5643	0.5643	1	1	3
GA-net	0.5643	0.5643	1	1	3
MOCD	0.5214	0.5214	0.8184	0.8184	5
MODPSO	0.5643	0.5634	1	0.9959	3

Table 4
The parameters of the signed networks. E^+ and E^- denote the positive and negative edges, respectively.

Network	#Vertex	#Edge	$\#E^+$	$\#E^-$
SPP	10	45	18	27
GGS	16	58	29	29
EGFR	329	779	515	264
Macrophage	678	1425	947	478
Yeast	690	1080	860	220
Ecoli	1461	3215	1879	1336

real community structures of the two synthetic signed networks as shown in Fig. 4.

When dealing with the two small synthetic networks, all the algorithms except MOCD perform similarly good. In order to further check the performance of the proposed algorithm, we will test the algorithm on real-world signed network in the next subsection.

4.4. Results on real-world signed networks

In this part, we will test the performance of DPSO and the comparison algorithms on six real-world signed networks. The networks are the Slovene Parliamentary Party network (SPP) (Kropivnik & Mrvar, 1996), the Gahuku-Gama Subtribes network (GGS) (Read, 1954), the epidermal growth factor receptor pathway network (EGFR) (Oda, Matsuoka, Funahashi, & Kitano, 2005), the macrophage network (Macrophage) (Oda et al., 2004), the yeast network (Yeast) (Milo et al., 2002), and the gene regulatory network of the Escherichia coli (Ecoli, http://regulondb.ccg.unam.mx, version 6.3) (Shen-Orr, Milo, Mangan, & Alon, 2002). The parameters of each network are given in Table 4.

For each social network, we ran each of the algorithms 30 times. We record the maximal and averaged values of the *SQ* and the *NMI* indexes. We also record the number of communities found by the algorithms for each network. The statistical results are summarized in Tables 5–8.

Table 5
Experimental results over 30 runs on the SPP signed network.

Algorithm	SQ_{max}	SQ_{avg}	NMI_{max}	NMI_{avg}	Cluster
DPSO	0.4547	0.4547	1	1	2
GA-net	0.4547	0.4547	1	1	2
MOCD	0.3634	0.3634	0.8047	0.8047	3
MODPSO	0.4547	0.4532	1	0.9949	2

Table 6
Experimental results over 30 runs on the GGS signed network.

Algorithm	SQ_{max}	SQ_{avg}	NMI_{max}	NMI_{avg}	Cluster
DPSO	0.4310	0.4310	1	1	3
GA-net	0.4310	0.4310	1	1	3
MOCD	0.3511	0.3375	0.8482	0.8262	5
MODPSO	0.4310	0.4310	1	1	3

Table 7
Experimental results over 30 runs on the EGFR and the Macrophage networks.

Network	EGFR			Macrophage		
Index	Q_{max}	Q_{avg}	Cluster	Q_{max}	Q_{avg}	Cluster
DPSO	0.3301	0.3215	44	0.3525	0.3278	116
GA-net	0.3247	0.3158	50	0.3478	0.3412	105
MOCD	0.3112	0.3041	60	0.3380	0.3308	139
MODPSO	0.2823	0.2798	20	0.3175	0.3101	54

Table 8
Experimental results over 30 runs on the Yeast and the Ecoli networks.

Network	Yeast			Ecoli		
Index	Q_{max}	Q_{avg}	Cluster	Q_{max}	Q_{avg}	Cluster
DPSO	0.6146	0.6097	110	0.4169	0.4138	392
GA-net	0.3692	0.3586	19	0.2618	0.2532	22
MOCD	0.3300	0.3119	27	0.2244	0.2070	31
MODPSO	0.6007	0.5988	108	0.3626	0.3348	393

Tables 5 and 6 show the experimental results of the algorithms on the SPP and the GGS signed networks. Our proposed algorithm can correctly recognize the true partitions of the two networks. The

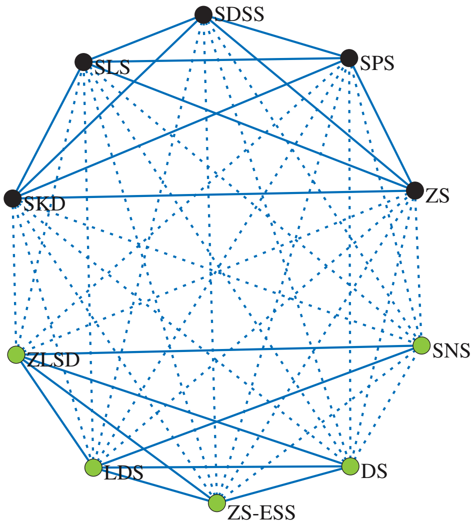


Fig. 5. The community structure of the SPP network detected by DPSO.

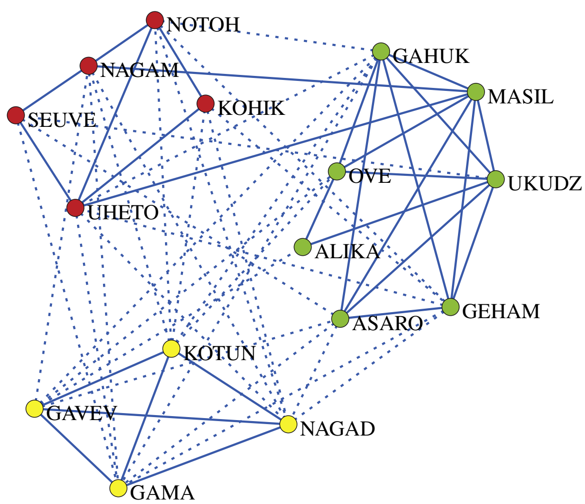


Fig. 6. The community structure of the GGS network detected by DPSO.

detected true partitions of the two networks are shown in Figs. 5 and 6.

In order to test the convergence of our proposed algorithm, for each network we run our algorithms 30 times and in each run we record the objective function values over each algorithm iteration. We then draw the convergence trajectory of the averaged objective function values over the algorithm iterations and we compare it with that of GA-net.

Fig. 7 exhibits the convergence trajectories of DPSO and GA-net when experimenting on the SPP and the GGS networks. Though DPSO and GA-net obtain the same objective values, from the convergence curves we may see that DPSO converges faster. The experiments on the two small networks demonstrate the effectiveness of the proposed algorithm. Since the networks are too small, it is very easy for the algorithms to get the optimal solutions. To further check the performance of the algorithms, we will test them on four big signed networks. Tables 7 and 8 summarize the experimental results on the four big signed networks.

It can be seen from the tables that our proposed algorithm can get higher objective function values. GA-net optimizes the same objective as DPSO, but from the experiments we clearly notice that DPSO can obtain higher objective values. To check the stabilities of the algorithms, we show the box plot of the signed modularity values obtained by the algorithms over 30 runs on the four big networks. The box plot results are shown in Fig. 8.

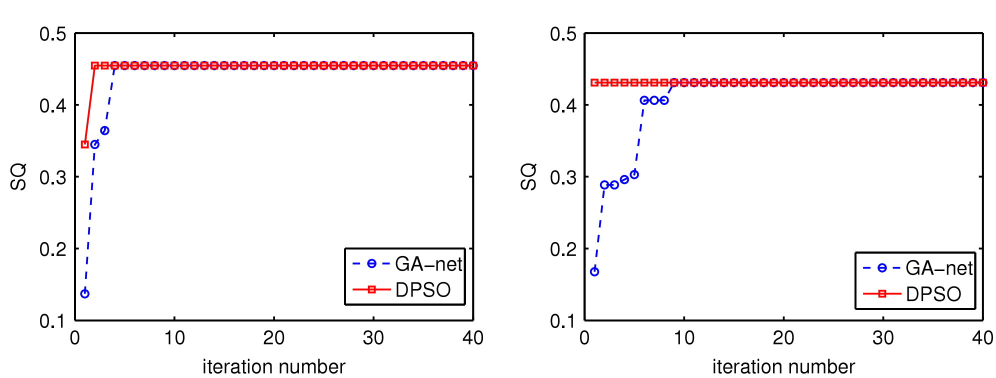
From the distribution of the box plots data we may notice that DPSO is very steady when testing on the Yeast and the Ecoli networks. In general, DPSO is superior to the comparison algorithms. Through our experiments we also find that our proposed algorithm can converge faster. Fig. 9 exhibits the convergence performance obtained with 30 independent runs of DPSO and GA-net.

From the convergence curves drawn in Fig. 9 we can see that it is obvious that DPSO converges very fast. Our proposed algorithm makes use of the network linkage information to guide the search of the particle. Each component of the algorithm has been carefully designed. We also implement the heuristic based method to initialize the population with high diversity. All these merits make the proposed algorithm capable of handling large scale social networks. All the experiments demonstrate that the proposed method is effective.

5. Concluding remarks

In this paper, a discrete particle swarm optimization method designed for signed network community detection is proposed for the first time. In our proposed approach the particles' position and velocity have been redefined in discrete form. The definitions are straightforward and easy to realize. Based on the definitions, the particles' updating rules have also been redefined. The newly defined rules make full use of signed network prior knowledge which benefits the algorithm from fast convergence. A novel swarm initialization mechanism proposed in our previous work is adopted to speed up convergence.

To validate the performances of the proposed method, extensive experiments on both synthetic and real-world signed



(a) SPP. (b) GGS.

Fig. 7. The convergence comparisons between DPSO and GA-net on the SPP and the GGS networks.

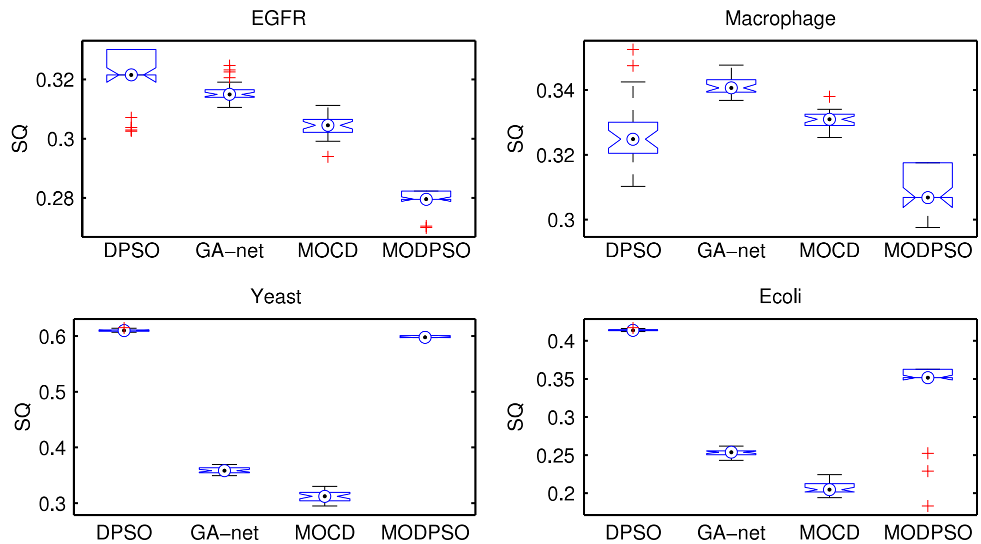
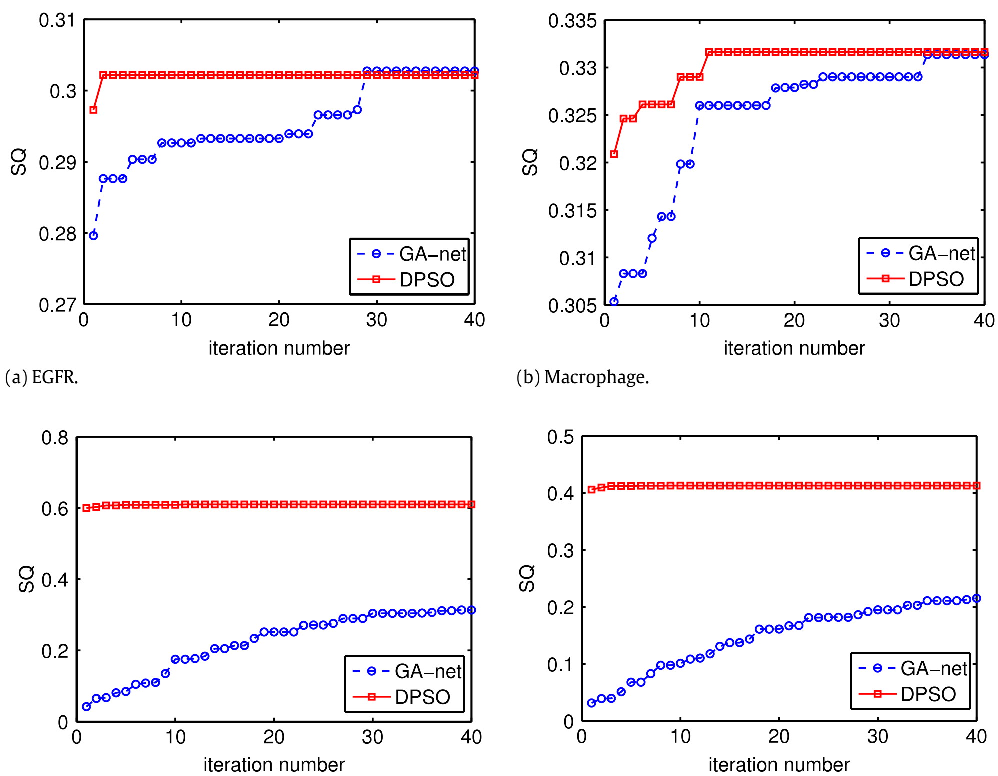


Fig. 8. Box plots of the signed modularity values obtained by the algorithms over 30 runs on the four big signed networks.



(a) EGFR. (b) Macrophage.

(c) Yeast. (d) Ecoli.

Fig. 9. The convergence comparisons between DPSO and GA-net over 30 independent runs.

networks are implemented. We also compare our method with three state-of-the-art approaches. When experimenting on the two synthetic signed networks, our proposed algorithm can correctly figure out the true community structures of the two networks. To further check its performance, we test our algorithm on two small and four big real-world signed networks. The true community structures of the two small networks are known while the rest four are unknown. On the two small networks, our algorithm can totally discover the truth of the networks. Through our experiments we find that our algorithm is competitive to the comparison algorithms on the small networks while on the four big networks it outperforms the rest in terms of the signed modularity values and the convergence time.

All the experiments demonstrate that the proposed algorithm for signed network community detection is effective and promising. However, there is still room to improve. The proposed algorithm actually provides a discrete particle swarm optimization paradigm for the clustering problem. Thus, it is believed that the proposed method can be applied to other domains. Moreover, since PSO is a stochastic searching algorithm, some problem-specific local search strategies can be designed to enhance the searching ability of the algorithm. Our future work will focus on more in-depth analysis of discrete PSO and its applications. Such analysis is expected to shed light on how discrete PSO can work better.

Acknowledgments

The authors wish to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality. This work was supported by the National Natural Science Foundation of China (Grant No. 61273317), the National Top Youth Talents Program of China, the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20130203110011) and the Fundamental Research Fund for the Central Universities (Grant No. K50510020001).

References

Albert, R., Jeong, H., & Barabasi, A. L. (1999). Internet-diameter of the world-wide web. *Nature*, *401*(6749), 130–131.

Al-kazemi, B., & Mohan, C. K. (2006). Multi-phase discrete particle swarm optimization. In *Information proceedings with evolutionary computation* (pp. 306–326). Springer.

Boccaletti, S., Ivanchenko, M., Latora, V., Pluchino, A., & Rapisarda, A. (2007). Detecting complex network modularity by dynamical clustering. *Physical Review E*, *75*, 045102.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., & Wagner, D. (2006). Maximizing modularity is hard. ArXiv preprint.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., & Wiener, J. (2000). Graph structure in the web. *Computer Networks*, *33*(1–6), 309–320.

Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, *28*(2), 15–21.

Chen, W. N., Zhang, J., Chung, H. S. H., Zhong, W. L., Wu, W. G., & Shi, Y. H. (2010). A novel set-based particle swarm optimization method for discrete optimization problems. *IEEE Transactions on Evolutionary Computation*, *14*, 278–300.

Clerc, M., & Kennedy, J. (2002). The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, *6*(1), 58–73.

Cook, D. J., & Holder, L. B. (2006). *Mining graph data*. John Wiley & Sons.

Danon, L., Diaz-Guilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Physics*.

Doreian, P. (2008). A multiple indicator approach to blockmodeling signed networks. *Social Networks*, *30*(3), 247–258.

Doreian, P., & Mrvar, A. (1996). A partitioning approach to structural balance. *Social Networks*, *18*(2), 149–168.

Duch, J., & Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical Review E*, *72*, 027104.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, *486*(3), 75–174.

Gangemi, A., Presutti, V., & Reforgiato Recupero, D. (2014). Frame-based detection of opinion holders and topics: a model and a tool. *IEEE Computational Intelligence Magazine*, *9*(1), 20–30.

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(12), 7821–7826.

Gómez, S., Jensen, P., & Arenas, A. (2009). Analysis of community structure in networks of correlated data. *Physical Review E*, *80*, 016114.

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

Gong, M., Cai, Q., Li, Y., & Ma, J. (2012). An improved memetic algorithm for community detection in complex networks. In *Proceeding of 2012 IEEE congress on evolutionary computation* (pp. 1–8).

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

Guimerò, R., Sales-Pardo, M., & Anaral, L. A. N. (2004). Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, *70*, 025101.

Hughes, B. D. (1995). *Random walks and random environments: random walks*. USA: Oxford University Press.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of 1995 IEEE international conference on neural networks* (pp. 1942–1948) *Vol. 4*.

Kennedy, J., & Eberhart, R. (1997). A discrete binary version of the particle swarm algorithm. In *Proceedings of 1997 IEEE international conference on systems, man, and cybernetics* (pp. 4104–4108) *Vol. 5*.

Kiranyaz, S., Ince, T., Yildirim, A., & Gabbouj, M. (2009). Evolutionary artificial neural networks by multi-dimensional particle swarm optimization. *Neural Networks*, *22*(10), 1448–1462.

Kleinberg, J. M. (2007). Challenges in mining social network data: processes, privacy, and paradoxes. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 4–5).

Kropivnik, S., & Mrvar, A. (1996). An analysis of the slovene parliamentary parties network. In A. Ferligoj, & A. Kramberger (Eds.), *Developments in statistics and methodology* (pp. 209–216).

Lau, R., Xia, Y., & Ye, Y. (2014). A probabilistic generative model for mining cybercriminal networks from online social media. *IEEE Computational Intelligence Magazine*, *9*(1), 31–43.

Liang, J., Qin, A., Suganthan, P., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, *10*(3), 281–295.

Liao, C. J., Tseng, C. T., & Luarn, P. (2007). A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers & Operations Research*, *34*(10), 3099–3111.

Ma, L., Gong, M., Liu, J., Cai, Q., & Jiao, L. (2014). Multi-level learning based memetic algorithm for community detection. *Applied Soft Computing*, *19*, 121–133.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., & Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, *298*(5594), 824–827.

Mitrovic, M., & Tadic, B. (2009). Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities. *Physical Review E*, *80*, 026123.

Newman, M. E. J. (2001). The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(2), 404–409.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, *69*, 066133.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(23), 8577–8582.

Oda, K., Kimura, T., Matsuoka, Y., Funahashi, A., Muramatsu, M., & Kitano, H. (2004). Molecular interaction map of a macrophage. *AfCS Research Reports*, *2*(14), 1–12.

Oda, K., Matsuoka, Y., Funahashi, A., & Kitano, H. (2005). A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology*, *1*(1).

Pampara, G., Franken, N., & Engelbrecht, A. (2005). Combining particle swarm optimisation with angle modulation to solve binary problems. In *Proceedings of 2005 IEEE congress on evolutionary computation* (pp. 89–96) *Vol. 1*.

Pizzuti, C. (2008). GA-Net: a genetic algorithm for community detection in social networks. In *Parallel problem solving from nature (PPSN). Vol. 5199* (pp. 1081–1090).

Pizzuti, C. (2012). A multiobjective genetic algorithm to find communities in complex networks. *IEEE Transactions on Evolutionary Computation*, *16*(3), 418–430.

Poria, S., Gelbukh, A., Hussain, A., Howard, N., Das, D., & Bandyopadhyay, S. (2013). Enhanced senticnet with affective labels for concept-based opinion mining. *IEEE Intelligent Systems*, *28*(2), 31–38.

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.

Read, K. E. (1954). Cultures of the central highlands, new guinea. *Southwestern Journal of Anthropology*, *10*(1), 1–43.

Reichardt, J., & Bornholdt, S. (2004). Detecting fuzzy community structures in complex networks with a potts model. *Physical Review Letters*, *93*(21), 218701.

Reichardt, J., & White, D. R. (2007). Role models for complex networks. *European Physical Journal B*, *60*(2), 217–224.

Salman, A., Ahmad, I., & Al-Madani, S. (2002). Particle swarm optimization for task assignment problem. *Microprocessors and Microsystems*, *26*(8), 363–371.

Sha, D. Y., & Hsu, C. Y. (2006). A hybrid particle swarm optimization for job shop scheduling problem. *Computers & Industrial Engineering*, *51*(4), 791–808.

Sharafi, M., & ELMekkawy, T. Y. (2014). Multi-objective optimal design of hybrid renewable energy systems using pso-simulation based approach. *Renewable Energy*, *68*(0), 67–79.

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, *31*(1), 64–68.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of 1998 IEEE congress on evolutionary computation* (pp. 69–73).

Shi, Y., & Eberhart, R. (1999). Empirical study of particle swarm optimization. In *Proceedings of 1999 congress on evolutionary computation* (p. 3) *Vol. 3*.

Shi, Y., & Eberhart, R. (2001). Fuzzy adaptive particle swarm optimization. In *Proceedings of 2001 IEEE congress on evolutionary computation* (pp. 101–106) *Vol. 1*.

Shi, C., Yan, Z. Y., Cai, Y. N., & Wu, B. (2012). Multi-objective community detection in complex networks. *Applied Soft Computing*, *12*(2), 850–859.

Ting, T. O., Shi, Y., Cheng, S., & Lee, S. (2012). Exponential inertia weight for particle swarm optimization. In *Advances in swarm intelligence* (pp. 83–90).
Traag, V., & Bruggeman, J. (2009). Community detection in networks with positive and negative links. *Physical Review E*, *80*(3), 036115.
van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *8*(3), 225–239.
Wu, F.-Y. (1982). The potts model. *Reviews of Modern Physics*, *54*(1), 235.
Wu, F., & Huberman, B. A. (2004). Finding communities in linear time: a physics approach. *European Physical Journal B*, *38*(2), 331–338.
Xu, R., Venayagamoorthy, G. K., & Wunsch, D. C., II (2007). Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. *Neural Networks*, *20*(8), 917–927.
Yang, B., Cheung, W. K., & Liu, J. M. (2007). Community mining from signed social networks. *The IEEE Transactions on Knowledge and Data Engineering*, *19*(10), 1333–1348.
Zhou, Z., & Shi, Y. (2011). Inertia weight adaption in particle swarm optimization algorithm. In *Advances in swarm intelligence* (pp. 71–79).
Ziv, E., Middendorf, M., & Wiggins, C. H. (2005). Information-theoretic approach to network modularity. *Physical Review E*, *71*, 046117.