Appl Intell (2016) 44:621–633
DOI 10.1007/s10489-015-0716-4

A novel multiobjective particle swarm optimization algorithm for signed network community detection

Zhaoxing Li[1,2] · Lile He[1] · Yunrui Li[3]

Published online: 26 October 2015

© Springer Science+Business Media New York 2015

Abstract Signed graphs or networks are effective models for analyzing complex social systems. Community detection from signed networks has received enormous attention from diverse fields. In this paper, the signed network community detection problem is addressed from the viewpoint of evolutionary computation. A multiobjective optimization model based on link density is newly proposed for the community detection problem. A novel multiobjective particle swarm optimization algorithm is put forward to solve the proposed optimization model. Each single run of the proposed algorithm can produce a set of evenly distributed Pareto solutions each of which represents a network community structure. To check the performance of the proposed algorithm, extensive experiments on synthetic and real-world signed networks are carried out. Comparisons against several state-of-the-art approaches for signed network community detection are carried out. The experiments demonstrate that the proposed optimization model and the algorithm are promising for community detection from signed networks.

✉ Zhaoxing Li
abde0309@126.com

[1] School of Mechanical and Electrical Engineering, Xi’an University of Architecture and Technology, Xi’an, Shaanxi Province 710055, China

[2] College of Information Engineering, Yulin University, Yulin, Shaanxi Province 719000, China

[3] Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA

Keywords Signed network · Community detection · Particle swarm optimization · Multiobjective optimization

1 Introduction

The network is ubiquitous. Network facilitates us with enormous convenience to better communicate with others. Network is changing our daily life unprecedentedly. The term “network” here mainly refers to the mobile communication networks. Besides the communication networks, there are a lot of other networks in real life, such as the power or the traffic transportation networks, the metabolic networks, etc. To do research on these complex networks is of great importance from the perspective of theoretical analysis and real applications.

A direct way to analyze complex networks is to represent networks by graphs. A graph is comprised of a set of nodes and edges. The nodes represent the objects that consist of the networks and the edges denote the relations among the objects. By analyzing the characteristics of a graph one can get to know the properties of a network. Networks have many famous properties, such as the small-world property [26], the scale-free property [2], etc., and among them the network community structure property [10] has been proved to be an eminent one. The network community property is in accordance with the Chinese old saying that “birds of a feather flock together”. A feather is like a community. In graph language, a community is referred to as a subgraph which holds the condition that within a community the similarities between the nodes are high while between different communities the similarities are low.

The community structure is very important for many networks. Consequently, to discover the community structures of complex networks has aroused great interests of scholars. So far, a large amount of methods have been proposed to detect the community structures of networks [8, 15, 23, 28]. One of the landmark contribution should owe to Girvan and Newman for their work in [10]. In their work, the modularity index has been put forward. The modularity index has been widely used to evaluate the goodness of a network partition. Following this work, many methods based on the modularity index have been cranked out [8]. The essence of community discovery is a clustering problem and a clustering problem can be effectively solved by heuristic optimization methods. With respect to this, a massive amount of creative works based on heuristic optimization algorithms have been done [5, 8]. To solve the signed network community detection problem, Cai et al. in [4] have proposed a particle swarm optimization algorithm which aims to maximize the signed modularity index [11]. However, the modularity based index exists the resolution limitation [9], i.e., by purely optimizing the modularity based index one cannot discover small communities whose sizes are smaller than a scale which depends on the total size of the network and on the degree of inter connectedness of the community. Apart from this, each sigle run of the algorithm in [4] can only produce one network community structure which is inconvenient for the decision maker. To avoid these two drawbacks, Gong et al. in [12] have put forward a state-of-the-art multiobjective discrete particle swarm optimization (MODPSO) algorithm for complex network clustering. Although the MODPSO algorithm is quite promising for community discovery, its search ability still needs improvement from the viewpoint of multiobjective optimization. Besides, from the viewpoint of network community discovery, the established optimization model also can be improved.

In this paper, to better solve the problem of community discovery from signed networks, a novel algorithm based on multiobject particle swarm optimization is newly proposed. The main contributions of this paper are as follows:

1. A new multiobjective optimization model is suggested for the signed network community discovery problem. The newly proposed model is based on the link density of a node. By optimizing the model one can ensure that both the positive link density within a community and the negative link density between different communities are big, which is in accordance with the signed community property.
2. A novel multiobject particle swarm optimization (MOPSO) algorithm based on the decomposition strategy is developed to solve the established multiobjective optimization model. In the newly developed algorithm, an efficient subproblem update strategy is proposed to enhance population diversity.
3. Extensive experiments compared against several state-of-the-art signed network community discovery algorithms are carried out. The experiments demonstrate that the proposed optimization model is effective and the devised subproblem update strategy does improve the performance of the proposed algorithm for signed network community discovery.

The rest of the paper is organized as follows. Section 2 gives the related backgrounds including the statement of the network community discovery problem and the brief introduction of MOPSO. Section 3 presents the proposed algorithm for network clustering in detail. Section 4 shows the experimental studies of the proposed method, and the conclusions are finally drawn in Section 5.

2 Related backgrounds

2.1 Problem description

A signed network is normally represented by a signed graph that is comprised of a set of nodes and edges. The edges contain two types, the positive edges and the negative edges. The task of signed network community detection is to divide a signed network into different clusters based on certain principles. Each cluster is called a community. Fig. 1 shows an example of a signed network with 9 nodes and 16 edges.

Currently, there is no uniform definition of what a community is. In academic domain, a community is regarded as a subset of a network that holds the condition that the similarities within a community are big while the similarities between different communities are small.

Gong et al. in [12] gave a condition that a signed community should satisfy. Given a signed network modeled as

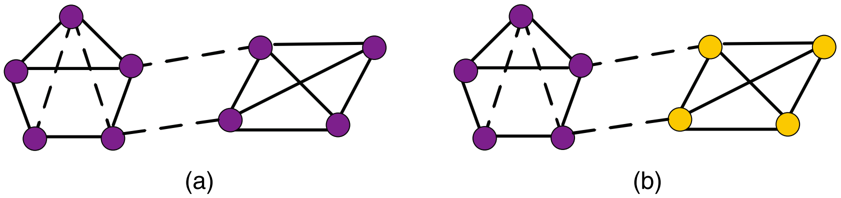


Fig. 1 A graphical illustration of **a** an example of a signed network, **b** an example of the corresponding community structure

$G = (V, PL, NL)$, where V is the set of nodes and PL and NL are the sets of positive and negative links, respectively. Let A be the adjacency matrix of G and l_{ij} be the link between nodes i and j.

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

$$\forall i \in S, \quad (d_i^+)^{in} > (d_i^-)^{in} \tag{1}$$

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

$$\begin{cases} \sum_{i \in S} (d_i^+)^{in} > \sum_{i \in S} (d_i^+)^{out} \\ \sum_{i \in S} (d_i^-)^{out} > \sum_{i \in S} (d_i^-)^{in} \end{cases} \tag{2}$$

The above condition indicates that, 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 and the negative links between different communities are all dense.

As can be seen from Fig. 1 that the task of community discovery from a signed network is to separate the whole network into many communities. However, how to evaluate the goodness of a separation is still an open issue.

For a signed network, in order to give a quantitative standard to a signed community structure, Gómez et al. in [11] presented a reformulation of the modularity metric which was originally proposed by Girvan and Newman [10]. The reformulated index is called the signed modularity (*SQ*) which reads:

$$SQ = \frac{1}{2w^+ + 2w^-} \sum_{i,j} \left(w_{ij} - \left(\frac{w_i^+ w_j^+}{2w^+} - \frac{w_i^- w_j^-}{2w^-} \right) \right) (\delta(i,j) \tag{3}$$

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 nodes 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 is, the better the separation of the community structure is.

2.2 Multiobjective particle swarm optimization

Particle swarm optimization (PSO) is a well known stochastic searching algorithm. PSO has been widely applied to solve a wide range of hard optimization problems [18]. PSO is an eminent optimization technique among the swarm intelligence algorithms which are originated from the social behaviors such as fish schooling and birds flocking. It was first proposed by Eberhart and Kennedy [13] in 1995.

PSO is also a population based algorithm. It optimizes a problem by having a swarm of individuals each of which is called a particle. Each particle has a position and a velocity vector. The position represents a candidate solution to the optimization problem. The velocity denotes the tendency for one particle to change its current position. The flight status of a particle is updated by simple rules.

Given that the size of the particle swarm is *pop* and the dimension of the search space is n. Let $\boldsymbol{V}_i = \{v_{i1}, v_{i2}, \ldots, v_{in}\}$ and $\boldsymbol{X}_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ be the ith ($i = 1, 2, ..., pop$) particle' velocity and position vectors, respectively. Then the rules for particle i to adjust its status are as the following:

$$\boldsymbol{V}_i \leftarrow \boldsymbol{V}_i + c_1 r_1 (\boldsymbol{P}_i - \boldsymbol{X}_i) + c_2 r_2 (\boldsymbol{G} - \boldsymbol{X}_i) \tag{4}$$

$$\boldsymbol{X}_i \leftarrow \boldsymbol{X}_i + \boldsymbol{V}_i \tag{5}$$

where $\boldsymbol{P}_i = \{p_{i1}, p_{i2}, \ldots, p_{in}\}$ is the ith particle's personal best position and $\boldsymbol{G} = \{g_1, g_2, \ldots, g_n\}$ is the best position of the swarm. Parameter c_1 and c_2 are the learning factors, and $r_1, r_2 \in [0, 1]$ are two random numbers.

It can be seen from the above descriptions that the canonical PSO is designed for continuous optimization problems. Many efforts have been made to extend the basic PSO into discrete contexts, and many discrete PSO (DPSO) algorithms have been proposed [1, 6, 14, 20]. In real applications, many optimization problems involve more than one optimization objective. These problems are the so called multiobjective optimization problems (MOPs). In order to solve MOPs, multiobjective PSO (MOPSO) algorithms have emerged. In the existing literatures, the framework of a general MOPSO is shown in Fig. 2.

```
Begin
    Initialize swarm
    Initialize leaders in an external archive
    Quality (leaders)
    g = 0
    While g < gmax
        For each particle
            Select leader
            Update position
            Mutation
            Evaluation
            Update pbest
        End For
        Update leaders in the external archive
        g++
    End While
    Report results in the external archive
End
```

Fig. 2 Pseudocode of a general MOPSO algorithm

Most of the existing MOPSOs apply some sort of mutation operator to promote diversity after performing the flight. It is obvious that the scheme of the basic PSO has to be modified if we want to apply PSO to MOPs, because for a MOP, we aim to find a set of different solutions. With respect to the leaders selection, the straight forward method is random selection, but it is not good to guide the flight. Another most simple approach is to adopt aggregating functions (i.e., weighted sums of the objectives) or approaches that optimize each objective separately. Other techniques based on the concept of Pareto optimality can also be utilized as a selection method, such as the density measure [7].

3 Proposed method for signed network community discovery

3.1 Framework of the proposed algorithm

Considering that the positive links within a signed community and the negative links between different communities should be dense, we separately model the two conditions as two objectives and consequently, a bi-objective optimization model is newly established in this paper. Let $C = \{c_1, c_2, ..., c_k\}$ be a network partition where k is the number of communities. The two objectives are shown as the following:

$$\min\ F(\cdot) \begin{cases} f_1 = -\sum_{i=1}^{k} \frac{L^+(c_i,c_i)-L^+(c_i,\overline{c_i})}{|c_i|} \\ f_2 = -\sum_{i=1}^{k} \frac{L^-(c_i,\overline{c_i})-L^-(c_i,c_i)}{|c_i|} \end{cases} \tag{6}$$

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

In the proposed optimization model, to minimize f_1 we can maximize the positive links within a signed community while to minimize f_2 we can maximize the negative links between different communities.

In order to minimize these two objectives simultaneously, decomposition strategy is employed to decompose the optimization problem into many scalar optimization problems. Because in this paper the two objectives are discrete, what is more, it is hard to decide whether they are concave or not. Based on these, the Tchebycheff approach is adopted in our algorithm to decompose the multiobjective optimization problem into scalar optimization problems. The Tchebycheff decomposition technique is written as:

$$\begin{aligned} g^{te}(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z}^*) &= \max\nolimits_{1\le i\le k} w_i |f_i(\boldsymbol{x}) - z_i^*| \\ &\quad subject\ to\ \boldsymbol{x} \in \Omega \end{aligned} \tag{7}$$

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

Algorithm 1 Framework of the proposed PSOSCD algorithm

```
1.  Initialize Particles' positions: $X = \{x_1, x_2, ..., x_{pop}\}^T$.
2.  Repair $X$. // See Section 3.4 for more information.
3.  Initialize Particles' velocities: $V = \{v_1, v_2, ..., v_{pop}\}^T$.
4.  Initialize weighted vectors: $w = \{w_1, w_2, ..., w_{pop}\}^T$.
    Then figure out the neighborhood based on Euclidean
    distance between $w_i$ and $w_j$.
5.  Initialize the particles' historical best position vectors
    $Pbest$: $Pbest = \{p_1, p_2, ..., p_{pop}\}^T$.
6.  Initialize reference point $z^*$.
7.  Set $iter = 0$.
8.  For $i = 1, 2, ..., popsize$, do the following
9.  Randomly select one particle from the $pop$ sub-
    problems as the leader.
10. Calculate new velocity $v_i'$ according to Eq. 8.
11. Calculate new position $x_i'$ according to Eq. 9.
12. Repair $x_i'$.
13. Update the neighboring subproblems of $x_i$. // See
    Section 3.5 for more information.
14. Update reference point $z^*$.
15. Update personal best solution $p_i$.
16. If $iter < gmax$, then $iter++$ and go to step 8,
    otherwise, stop the algorithm and output.
```

Algorithm 1 shows the framework of the proposed algorithm (denoted by PSOSCD) for signed network community discovery. In step 2 and step 12, we have developed a position repair operation which aims to save computational time. The detailed operations are shown in Section 3.4. In step 11, in order to better preserve population diversity, in this paper we have devised a novel replacement strategy to update the neighboring subproblems of x_i. The detailed operations are shown in Section 3.5.

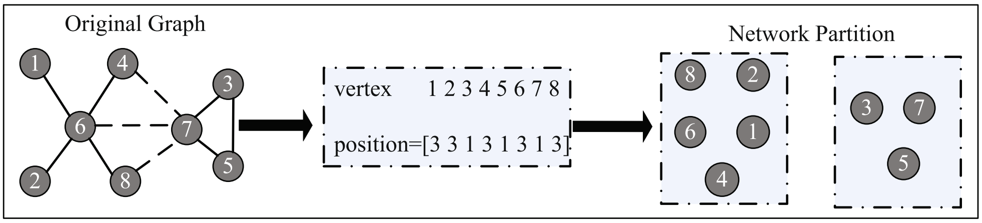


Fig. 3 A schematic illustration of the string based particle representation schema

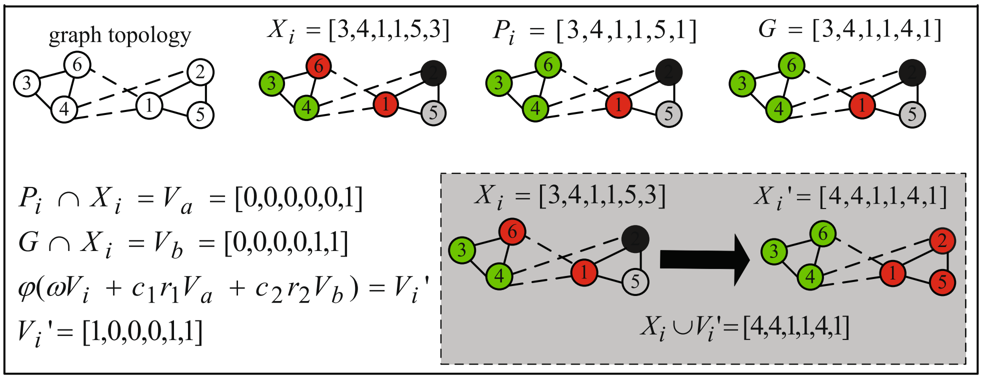

graph topology
X_i = [3,4,1,1,5,3]
P_i = [3,4,1,1,5,1]
G = [3,4,1,1,4,1]
$P_i \cap X_i = V_a$ = [0,0,0,0,0,1]
$G \cap X_i = V_b$ = [0,0,0,0,1,1]
$\varphi(\omega V_i + c_1 r_1 V_a + c_2 r_2 V_b) = V_i'$
V_i'= [1,0,0,0,1,1]
X_i = [3,4,1,1,5,3]
X_i'= [4,4,1,1,4,1]
$X_i \cup V_i'$=[4,4,1,1,4,1]
1
2
3
4
5
6

Fig. 4 A graphical illustration of the operations of the update rules

3.2 Particle coding/decoding scheme

A particle represents a solution to the optimization problem. The update rules in (4) and (5) are for the particles to reproduce offspring solutions. For the community detection problem, this paper adopts the string based coding scheme in which the position vector of a particle is an integer permutation while the veloity vector is a binary coded permutation. The adopted representation schema is shown in Fig. 3.

It can be seen from the figure that the string based coding scheme is direct and it is easy to decode. What most important is that this kind of scheme does not need to specific the community size in advance.

3.3 Particle update rules

From Fig. 3 we can notice that the (4) and (5) no longer fit for the community detection problem. In this paper we have redefined (4) and (5). The redefined rules are shown as follows:

$$\boldsymbol{V}_i \leftarrow \varphi(\omega \boldsymbol{V}_i + c_1 r_1 (\boldsymbol{P}_i \cap \boldsymbol{X}_i) + c_2 r_2 (\boldsymbol{G} \cap \boldsymbol{X}_i)) \tag{8}$$

$$\boldsymbol{X}_i \leftarrow \boldsymbol{X}_i \cup \boldsymbol{V}_i \tag{9}$$

In the above equations, the symbol "∩" is the XOR operator. The function $\varphi(t)$ is defined as follows:

$$\varphi(t) = \begin{cases} 1 \ if \ rand(0, 1) \leq 1/(1 + e^{-t}) \\ 0 \ if \ rand(0, 1) > 1/(1 + e^{-t}) \end{cases} \tag{10}$$

The operation rule for the symbol "∪" in (9) is defined in the following way:

$$\begin{cases} X_i \ominus V_i = X_i' = \{x_{i1}', x_{i2}', ..., x_{in}'\} \\ x_{ij}' = x_{ij} \ if \ v_i = 0 \\ x_{ij}' = \arg\max_r \sum_{k \in N_j} \delta(x_{ik}, r) \ if \ v_i = 1 \end{cases} \tag{11}$$

where N_j is the neighbor set of node j. If $a = b$, $\delta(a, b) = 1$, otherwise, 0.

Figure 4 graphically depicts the detailed operations of the update rules. It can be seen that the particle status update rules make use of the topology information of a signed network, which makes the proposed algorithm feasible for addressing the community discovery task.

3.4 Position repair

From the coding system described above one may notice a phenomenon that two different position vectors may correspond to the same network community structure. Figure 5 shows an example of this phenomenon.

In Fig. 5, $\boldsymbol{X}_i$ and $\boldsymbol{P}_i$ are the ith particle's current and historically best positions respectively. After decoding, $\boldsymbol{X}_i$ and $\boldsymbol{P}_i$ represent the same network community structure. Based on (8) and (9) we will get a non-zero vector $\boldsymbol{V}_i$, and $\boldsymbol{X}_i$ will be changed which takes time. In order to save computational time, we have devised a position repair operation whose pseudocodes are shown in Algorithm 2.

Algorithm 2 Pseudocodes of the proposed position repair operation

```
1. For a given position vector $X$, find the unique elements
   and store them in $U$.
2. for $i = 0;\ i < U.size();\ i++$
    • for $j = 0;\ j < X.size();\ j++$
        – if ($X[j] == U[i]$), then $X[j] = i + 1$.
    • end for
3. end for
```

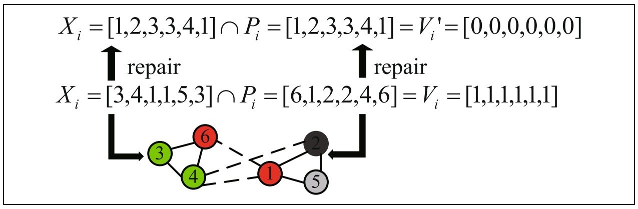

X_i =[1,2,3,3,4,1] ∩ P_i =[1,2,3,3,4,1] = V_i'=[0,0,0,0,0,0]
repair
repair
X_i =[3,4,1,1,5,3] ∩ P_i =[6,1,2,2,4,6] = V_i =[1,1,1,1,1,1]
1
2
3
4
5
6

Fig. 5 An example of two different position vectors that correspond to the same network community structure

Table 1 Meanings of the parameters in the extended LFR model for generating signed benchmark networks

Parameter	Meaning
n	number of nodes
k_{avg}	averaged degree of a node
k_{max}	maximum degree of a node
γ	exponent for the power law distribution of the node degree
β	exponent for the power law distribution of the community size
s_{min}	minimum community size
s_{max}	maximum community size
μ	mixing parameter
$p-$	fraction of negative edges within communities
$p+$	fraction of positive edges between different communities

As shown in Fig. 5, after we repair the position, $\boldsymbol{X}_i$ and $\boldsymbol{P}_i$ become the same and based on (8) a zero vector is obtained. Then there is no need to compute the new position vector and the computational time is saved.

3.5 Replacement operation

The replacement operation aims to replace the subproblems with better offspring solutions. In this paper, in order to better preserve population diversity, we have proposed a novel subproblems update strategy. Given that $\boldsymbol{x}_i^{t+1}$ is a newly generated solution, in our proposed update strategy, only T subproblems are updated. The pseudocode of the proposed strategy is shown in Algorithm 3.

Algorithm 3 The proposed subproblems update strategy

```
Set $B = \{1, ..., pop\}$. Set $temp = 0$.
For each index $j \in B$, if $g^{te}(\boldsymbol{x}'_i|\boldsymbol{\omega}_j, z^*) \le g^{te}(\boldsymbol{x}'_j|\boldsymbol{\omega}_j, z^*)$, then $temp = temp + 1$.
if $temp \le T$ then
    • For each index $j \in B$, if $g^{te}(\boldsymbol{x}'_i|\boldsymbol{\omega}_j, z^*) \le g^{te}(\boldsymbol{x}'_j|\boldsymbol{\omega}_j, z^*)$, then set $\boldsymbol{x}'_j = \boldsymbol{x}'_i$ and $F(\boldsymbol{x}'_j) = F(\boldsymbol{x}'_i)$.
else
    • for indexes $j \in B$ with $g^{te}(\boldsymbol{x}'_i|\boldsymbol{\omega}_j, z^*) \le g^{te}(\boldsymbol{x}'_j|\boldsymbol{\omega}_j, z^*)$ do
        – Rank the Euclidean distances between $F(\boldsymbol{x}'_i)$ and $F(\boldsymbol{x}'_j)$ with an ascend order.
        – Choose the former $T$ solutions, then set $\boldsymbol{x}'_j = \boldsymbol{x}'_i$ and $F(\boldsymbol{x}'_j) = F(\boldsymbol{x}'_i)$.
    • end for
end if
```

The parameter T cannot be to large. On one hand, if T is too big, it will cost a lot of computational time. On the other hand, since T subproblems will be replaced by the newly generated good individual, if T is too big, the population diversity will decrease immediately.

4 Experimental testing

4.1 Experimental settings

In this part, the proposed PSOSCD algorithm will be tested on both benchmark signed networks and real-world signed networks. Several state-of-the-art signed network community discovery methods have been selected to compare with the proposed approach. The proposed algorithm is coded in C++, and the experiments are carried out on an Inter(R) Celeron(R)M CPU 550 machine, 3.2GHz, with 4GB memory.

The comparison methods are the MODPSO algorithm [12] and the MOEA-SN algorithm [3]. For all the algorithms, the population size *pop*, and the maximal algorithm iteration number *gmax* are all set to 200. The crossover and mutation possibilities are set to 0.8 and 0.2, respectively. In PSOSCD, the learning factors c_1 and c_2 are both set to 1.494, and the inertia weight ω is set to 0.729.

In order to estimate the similarity between the true network partition and the discovered one, the widely used *Normalized Mutual Information* (*NMI*) metric [27] is adopted.

Table 2 Parameter settings for the 216 generated signed benchmark networks

Parameter	Value	Parameter	Value
n	500	s_{min}	10
k_{avg}	15	s_{max}	50
k_{max}	40	μ	[0:0.1:0.5]
γ	2	$p-$	[0:0.2:1.0]
β	1	$p+$	[0:0.2:1.0]

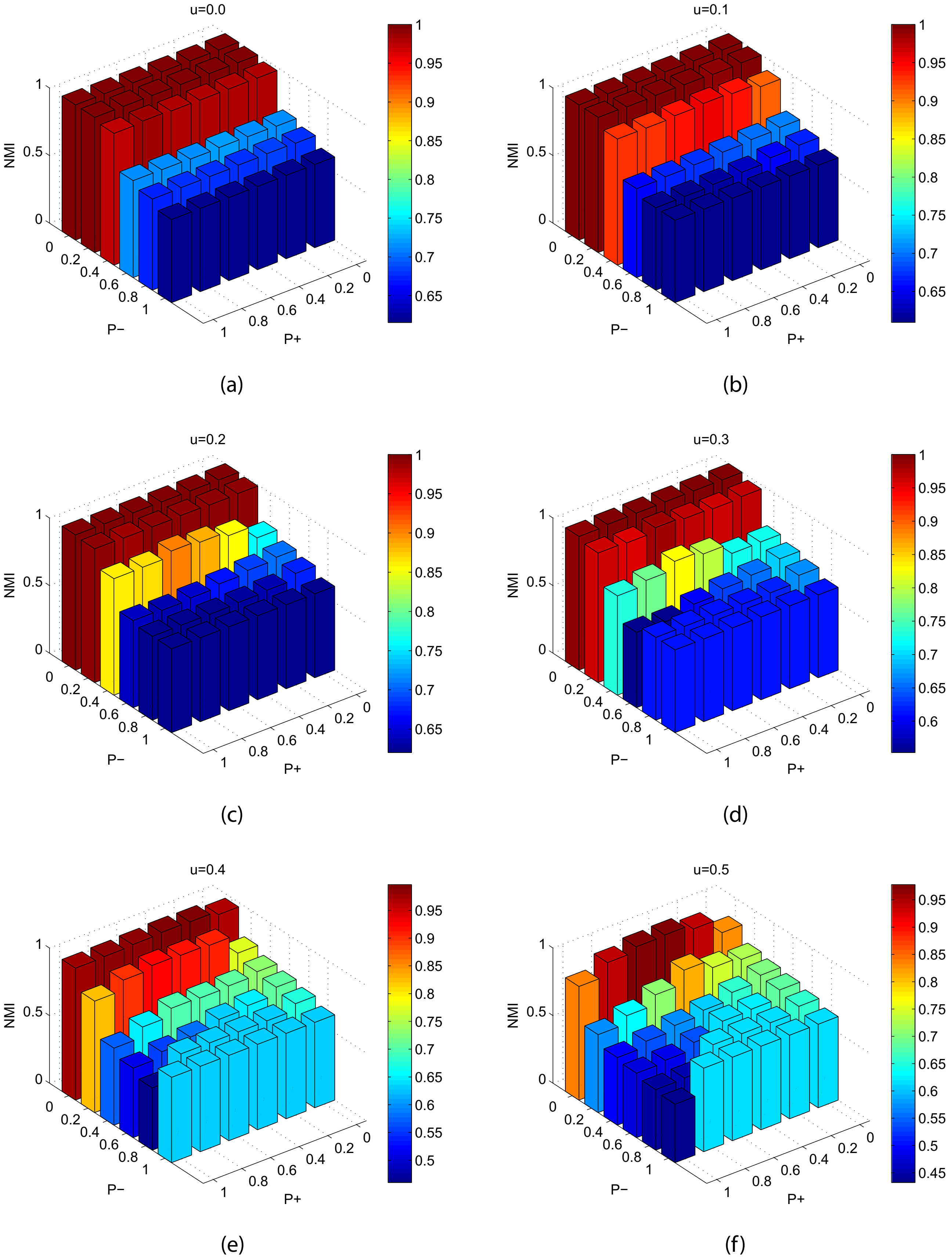


Fig. 6 Averaged *NMI* values for the 216 signed benchmark networks

Table 3 Statistics of the signed networks. E^+ and E^- denote the positive and negative edges, respectively

Network	#Vertex	#Edge	$\#E^+$	$\#E^-$	Ref.
SPP	10	45	18	27	[16]
GGS	16	58	29	29	[24]
EGFR	329	779	515	264	[22]
Macrophage	678	1425	947	478	[21]
Yeast	690	1080	860	220	[19]
Ecoli	1461	3215	1879	1336	[25]

Given that A and B are two partitions of a network, then the NMI between A and B is written as:

$$NMI = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij} log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{C_A} C_{i.} log(C_{i.}/N) + \sum_{j=1}^{C_B} C_{.j} log(C_{.j}/N)} \quad (12)$$

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). If $NMI(A, B) = 1$, then we say that A and B are the same.

4.2 Testing on synthetic signed networks

Lancichinetti et al. in [17] have cranked out the so called LFR model for generating unsigned benchmark networks for testing purpose. In this paper, this model has been extended to generate signed benchmark networks. The extended model is denoted by $SN(n, k_{avg}, k_{max}, \gamma, \beta, s_{min}, s_{max}, \mu, p-, p+)$. By tuning the parameters in SN we can control the structure of the generated network. The meanings of the parameters are listed in Table 1.

In this paper, we have generated 216 benchmark networks to test the performance of the proposed algorithm. Table 2 lists the parameters for the generated benchmark networks.

Because the real community structures of the benchmark networks are known, we use the *NMI* index to evaluate the performance of the algorithms. For the benchmark networks, we run the proposed algorithm and the comparison algorithms for 30 times and the averaged *NMI* values are recorded.

Figure 6 shows the averaged *NMI* values for the 216 signed benchmark networks. When the parameter μ increases, then the links within a community will be less, consequently the community structure becomes more and more vague, and it is harder and harder for an algorithm to detect the ground truths of the benchmark networks.

The structure of each signed network is mainly decided by the parameters μ, $p-$, and $p+$. If μ is bigger than 0.5, then there is hardly community structure in the network. Since we limit the range of μ from 0 to 0.5, consequently the network structure is mainly decided by the parameters $p-$ and $p+$. From Fig. 6 we can see that when $p-$ is small, the obtained *NMI* values are high, close to or equal 1. With $p-$ increases, the negative edges within a community is becoming denser, i.e., the community structure is becoming vague which makes it difficult for any methods to discover the community structures. Reflected in the figure, the *NMI*values decrease.

The experiments on the extended LFR signed benchmark networks indicate that the proposed algorithm works well on the benchmark networks. In the next subsection we are going to show the performances of the proposed algorithm on real-world signed networks.

Table 4 Hypervolumes of the PFs generated by the proposed algorithm with different settings of the parameter T

Network	HI	$T = 1$	$T = 2$	$T = 3$	$T = 4$	$T = 5$
SPP	Mean	1.4212	1.4212	1.4210	1.4212	1.4212
	Std	0	0	0.0013	0	0
GGS	Mean	1.3071	1.3146	1.3113	1.3004	1.2752
	Std	0.0065	0.0030	0.0167	0.0293	0.0402
EGFR	Mean	1.1321	1.1430	1.1385	1.1206	1.0732
	Std	0.0333	0.0318	0.0415	0.0527	0.0993
Macrophage	Mean	1.2038	1.2323	1.2387	1.2137	1.1882
	Std	0.0204	0.0202	0.0198	0.0354	0.0685
Yeast	Mean	1.1562	1.1782	1.1563	1.1338	1.1296
	Std	0.0338	0.0323	0.0323	0.0474	0.0502
Ecoli	Mean	1.1125	1.1343	1.1201	1.1127	1.0378
	Std	0.0398	0.0282	0.0337	0.0401	0.1105

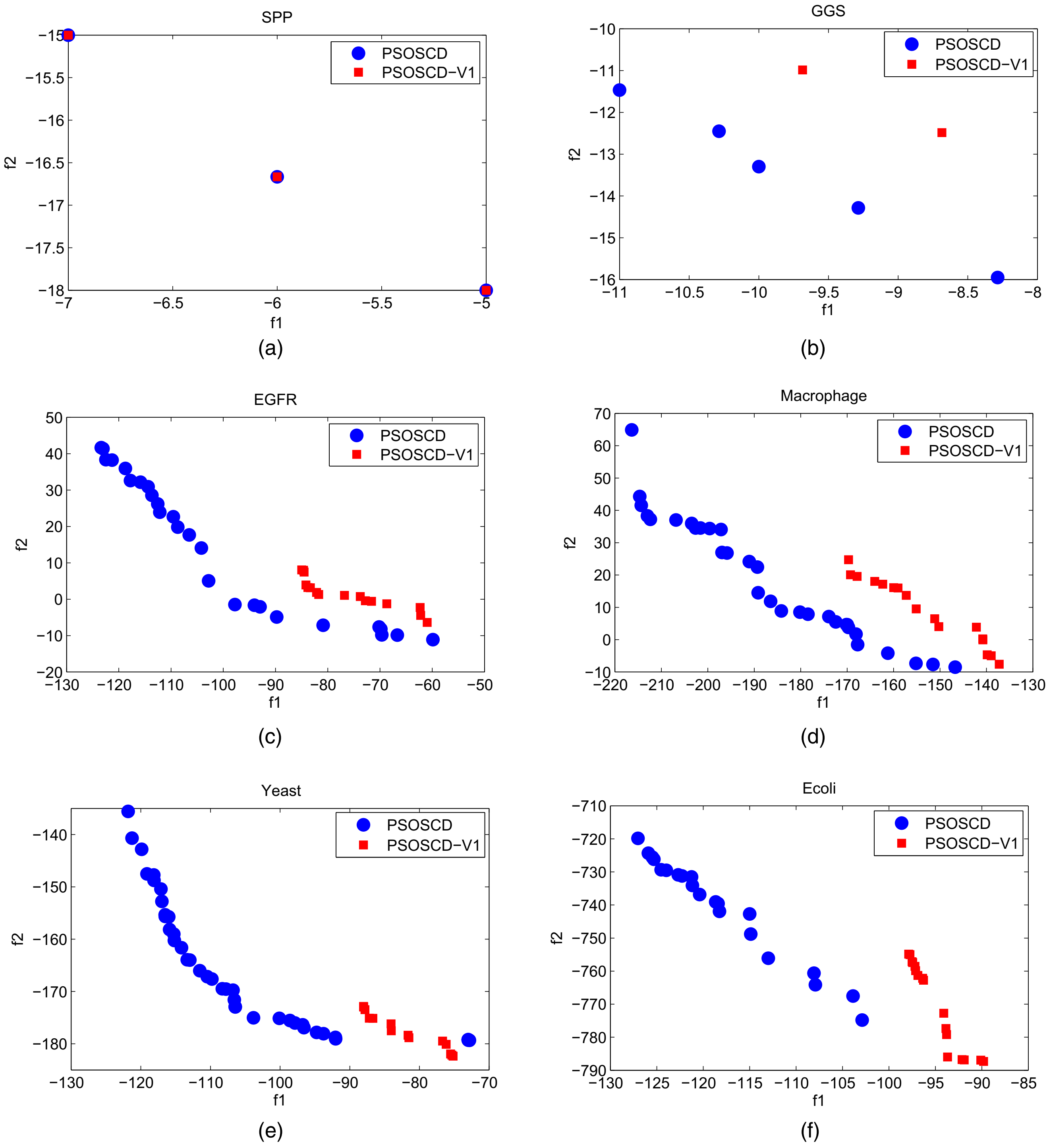


Fig. 7 Pareto fronts obtained by the algorithms with biggest hypervolumes

4.3 Testing on real-world signed networks

Six real-world signed networks have been employed to test the community discovery performances of the proposed algorithm and the comparison algorithms. The statistics of each network are given in Table 3.

In Table 3, the real community structures of the SPP and the GGS network are known. In our experiments,

Table 5 Statistical results over 30 runs on the signed networks with known ground truth

Data	Algorithm	NMI_{max}	NMI_{avg}	*Clusters*	SQ_{max}	SQ_{avg}
SPP	PSOSCD	1	1	2	0.4547	0.4547
	PSOSCD-v1	1	1	2	0.4547	0.4547
	MODPSO	1	0.9949	2	0.4547	0.4532
	MOEA-SN	1	0.9993	2	0.4547	0.4543
	PSOSCD	1	1	3	0.4310	0.4310
	PSOSCD-v1	1	1	3	0.4310	0.4310
GGS	MODPSO	1	1	3	0.4310	0.4310
	MOEA-SN	1	0.9955	3	0.4310	0.4303

each of the algorithm has been independently tested for 30 times.

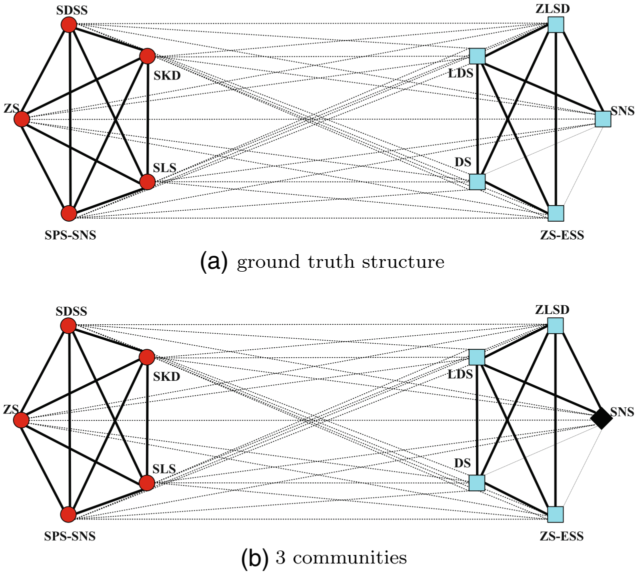
4.3.1 Impact of the parameter *T*

In our proposed subproblem update operation, there is a parameter *T* which affects the performance of our proposed algorithm. In this part we will test its impact.

Because different values of parameter *T* will lead to different Pareto fronts, since the true Pareto front for each network is unknown, in order to evaluate the goodness of the Pareto front yielded by the proposed algorithm, we adopt the hypervolume index (HI) [29] which calculates the area of the space covered by the Pareto front. The HI index, also known as the *S* metric or the Lebesgue measure, is calculated as follows:

$$\mathrm{HI}(PS, \boldsymbol{y}_{ref}) = \mathcal{L}\left(\cup_{\boldsymbol{y}\in PS}\left\{\boldsymbol{y}'|\boldsymbol{y} \prec \boldsymbol{y}' \prec \boldsymbol{y}_{ref}\right\}\right), \qquad (13)$$

where PS is the set of Pareto-optimal solutions; $\boldsymbol{y}_{ref} \in \mathbb{R}^m$ (m is the number of objectives) denotes the reference point that should be dominated by all Pareto-optimal solutions; $\mathcal{L}$ denotes the Lebesgue measure; and $\prec$ represents dominance.



(a) ground truth structure

(b) 3 communities

Fig. 8 Detected community structures of the SPP network

In our experiments, HI has been normalized and the reference point $\boldsymbol{y}_{ref}$ is set to (1.2, 1.2). We test our proposed algorithm with different configurations of the parameter *T*. We record the hypervolumes of the Pareto fronts when experimenting on the six real-world networks. The results over 30 runs are recorded in Table 4.

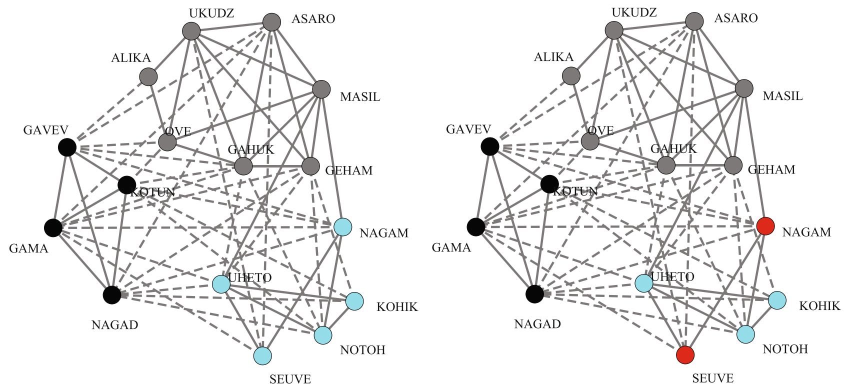
The parameter *T* determines the number of neighboring solutions which will be updated by a new offspring solution. If *T* is very big, it will be very time consuming and the whole population diversity will be lost because a good new offspring solution will update the majority of the solutions. If *T* is so small, the convergence speed will be slow because at each update step only a small number of solutions will be updated. The results in Table 4 suggest that $T = 2$ seems to be the best. Based on this, in all the experiments, we set $T = 2$ for our proposed algorithm.

4.3.2 Efficacy of the replacement operation

In this paper we have devised a new replacement operation to update the neighboring subproblems. In order to test the effectiveness of the proposed strategy, we compare the Pareto fronts obtained by PSOSCD and its variant PSOSCD-v1 which does not use the update strategy. We run both PSOSCD and PSOSCD-v1 for 30 times. The Pareto fronts with the biggest hypervolumes are shown in Fig. 7.

It can be seen from Fig. 7a that, for the SPP network, PSOSCD yields the same Pareto front as PSOSCD-v1 does. In our experiments we have found that each single run of these two algorithms always output the same results when experimenting on the SPP network, which indicates that the impact of the replacement operation is not significant. One reason that can explain this is because of the simple structure of the SPP network which makes the algorithm easy to find the optimal solutions.

From Fig. 7b to f we can clearly see that PSOSCD yields better Pareto fronts on the rest five real signed networks than



(a) ground truth structure

(b) 4 communities

Fig. 9 Detected community structures of the GGS network

PSOSCD-v1 does. The above experiments demonstrate that the proposed replacement operation can greatly enhance the population diversity.

4.3.3 Community discovery performance

In view of that the parameter T is determined and the efficacy of the replacement operation is validated, next step we will show the community discovery performance of our proposed algorithm.

For each network, we select the solution on the Pareto front that has the biggest signed modularity value as the final output of our proposed algorithm. Table 5 lists the statistical results when experimenting on the two small scale networks with known community structures.

From Table 5 we can see that, for the SPP and the GGS networks, all the algorithms perform similarly good, but the proposed algorithm performs the best. These two signed networks are small in size. It is easy for the algorithms to find the optimal solutions.

Figures 8 and 9 display the community structures obtained by our proposed algorithm when experimenting on the SPP and the GGS networks.

Each single run of the proposed algorithm can yield a set of solutions each of which denotes a network community structure. In Figs. 8 and 9 we respectively display two community structures of the corresponding network.

It can be seen that apart from the ground truth community structures, our proposed algorithm also discovered other interesting community structures. For the SPP network, it has discovered a structure with three communities in which the node SNS has been separated as an independent community. Because the node SNS has two negative linkages with the nodes in its original community, consequently, the discovered structure with three communities is meaningful. The similar phenomenon also happens to the GGS network. Apart from the ground truth structure, it also discovered other meaningful community structures which can facilitate intelligent decision making.

Table 6 lists the experimental results for the four networks with unknown ground truth. Because these four networks have no ground truth, we cannot compare the detected community structures with any reference structures. Besides, since the scales of these networks are big, it is hard to display their community structures. However,

Table 6 Statistical results over 30 runs on the signed networks with unknown ground truth

Index	Algorithm	EGFR	Macrophage	Yeast	Ecoli
Clusters	PSOSCD	83	73	89	123
	PSOSCD-v1	92	80	99	121
	MODPSO	93	85	103	125
	MOEA-SN	91	88	91	137
SQ_{max}	PSOSCD	0.2878	0.3244	0.6038	0.4032
	PSOSCD-v1	0.2731	0.2985	0.5879	0.3652
	MODPSO	0.2848	0.3028	0.5969	0.3979
	MOEA-SN	0.2753	0.3010	0.5838	0.3980
SQ_{avg}	PSOSCD	0.2835	0.3189	0.5979	0.3908
	PSOSCD-v1	0.2635	0.2787	0.5434	0.3301
	MODPSO	0.2791	0.2921	0.5741	0.3752
	MOEA-SN	0.2678	0.2738	0.5321	0.3696

from the viewpoint of the signed modularity we can notice that, the maximum values and the averaged values obtained by our proposed algorithm are higher than those obtained by the rest algorithms, which indicates that the discovered community structures are better than those discovered by the comparison algorithms.

All the experiments have proved the validity of the proposed algorithm. The proposed optimization model is effective for the signed network community discovery problem. The developed PSOSCD algorithm which makes use of the decomposition strategy solves the proposed model well. Meanwhile, the newly cranked out subproblems update strategy enhances the population diversity.

5 Conclusion

Community discovery from signed networks is an important task in complex network analytics. This paper modeled the task of signed network community discovery as a multiobjective optimization problem. Considering the very nature of a signed community, in this paper, a new multiobjective optimization model based on the link density was put forward. In order to solve the newly proposed model, a novel multiobjective particle swarm optimization algorithm was devised. The proposed algorithm divided the multiobjective optimization problem into many scalar optimization subproblems by using the Tchebycheff decomposition technique. Each subproblem was optimized through the particle swarm optimization technique. In order to enhance the diversity of the swarm, a novel subproblem update tactic was proposed. Each single run of the proposed algorithm can yield a set of equally good solutions each of which represents a certain community structure of the tested signed network. Extensive experiments compared against several state-of-the-art approaches on both synthetic and real-world signed networks demonstrated that the proposed optimization model was effective and the devised algorithm was promising for solving the signed network community detection problem.

In this paper, the proposed optimization model is promising for undirected and unweighted signed networks. However, in reality many signed networks are directed and have weights. What is more, the topologies of many signed networks may evolve with the time. In our future work, more efforts will be made to study the community detection problem under these contexts.

Acknowledgments This work was supported by the Science Project of Yulin City (Grant Nos. Gy13-15 and Ny13-10) and the Scientific Research Program of the Department of Education of Shaanxi Province (Grant No. 14JK1859).

References

1. Al-kazemi B, Mohan CK (2006) Multi-phase discrete particle swarm optimization. In: Information proceedings with evolutionary computation. Springer, pp 306–326
2. Barabási A-L, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512
3. Liu C, J L, Jiang Z (2014) A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. IEEE Trans Syst Man Cybern. doi:10.1109/TCYB.2014.2305974
4. Cai Q, Gong M, Shen B, Ma L, Jiao L (2014) Discrete particle swarm optimization for identifying community structures in signed social networks. Neural Netw 58 (P):4–13
5. Cai Q, Ma L, Gong M, Tian D (2015) A survey on network community detection based on evolutionary computation. Int J Bio-Inspired Comput 6(2):1–15
6. Chen WN, Zhang J, Chung HSH, Zhong WL, Wu WG, Shi YH (2010) A novel set-based particle swarm optimization method for discrete optimization problems. IEEE Trans Evol Comput 14:278–300
7. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
8. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3):75–174
9. Fortunato S, Barthelemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci 104(1):36–41
10. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826
11. Gómez S, Jensen P, Arenas A (2009) Analysis of community structure in networks of correlated data. Phys Rev E 80:016114
12. Gong M, Cai Q, Chen X, Ma L (2014) Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. IEEE Trans Evol Comput 18 (1):82–97
13. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of 1995 IEEE international conference on neural networks, vol 4, pp 1942–1948
14. 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, vol 5, pp 4104–4108
15. Kim J, Hwang I, Kim Y-H, Moon B-R (2011) Genetic approaches for graph partitioning: a survey. In: Proceedings of 13th annual conf. genetic evol. comput, pp 473–480
16. Kropivnik S, Mrvar A (1996) An analysis of the slovene parliamentary parties network. In: Ferligoj A, Kramberger A (eds) Developments in statistics and methodology, pp 209–216
17. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E 78(4)
18. Liang J, Qin A, Suganthan P, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10 (3):281–295
19. 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
20. Mitrovic M, Tadic B (2009) Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities. Phys Rev E 80:026123

21. Oda K, Kimura T, Matsuoka Y, Funahashi A, Muramatsu M, Kitano H (2004) Molecular interaction map of a macrophage. AfCS Res Rep 2(14):1–12
22. Oda K, Matsuoka Y, Funahashi A, Kitano H (2005) A comprehensive pathway map of epidermal growth factor receptor signaling. Mol Syst Biol 1(1)
23. Orman GK, Labatut V (2009) A comparison of community detection algorithms on artificial networks. In: Discovery science, pp 242–256
24. Read KE (1954) Cultures of the central highlands, new guinea. Southwest J Anthropol 10(1):1–43
25. Shen-Orr SS, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of escherichia coli. Nat Genet 31(1):64–68
26. Watts DJ, Strogatz SH (1998) Collective dynamics of ‘small-world’ networks. Nature 393(6684):440–442
27. Wu F, Huberman BA (2004) Finding communities in linear time: a physics approach. Eur Phys J B 38(2):331–338
28. Xie J, Kelley S, Szymanski BK (2013) Overlapping community detection in networks: The state-of-the-art and comparative study. ACM Comput Surv (CSUR) 45(4):43
29. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Parallel problem solving from nature. Springer, pp 292–301

Zhaoxing Li received the M.S. degree in software engineering from Xidian University, Xi’an, China, in 2010. Now he is working towards the Ph.D. degree in Mechanical design and theory at the School of Mechanical and Electrical Engineering, Xi’an University of Architecture and Technology, Xi’an, China. His current research interests are in the area of mechanical design, computational intelligence, and complex network analytics.