

Multiobjective local search for community detection in networks

Yalan Zhou¹ · Jiahai Wang² · Ningbo Luo² · Zizhen Zhang³

Published online: 8 May 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract Detecting communities is of great importance in the study of complex networks. In this study, the community detection problem is formulated as a multiobjective optimization problem; then a local search-based multiobjective optimization algorithm is proposed. In the proposed algorithm, different objectivewise local searches are designed for different objectives. These simple but effective local searches cooperate to simultaneously optimize two objectives. Extensive experiments on both synthetic and real-world networks show that the proposed algorithm obtains better or competitive results compared with existing state-of-the-art algorithms.

Keywords Multiobjective local search · Multiobjective optimization · Community detection

1 Introduction

Community detection in complex networks is a challenging problem. The aim of community detection algorithms is to discover the division of a network into groups of nodes, called community structures. The characteristic of commu-

nity structure is denser internal connectivity than external connectivity. Community structures exist in various fields, such as biological networks, science and technology networks (Castiglione et al. 2015; Esposito et al. 2013; Li et al. 2010), and social networks (Carullo et al. 2015). In recent years, community detection in networks has attracted a lot of attention and has been widely used in many different domains, such as terrorist organization identification, protein function prediction, and public opinion analysis.

A number of community detection approaches have been proposed. Extensive and detailed review of community detection approaches in complex networks can be found in a recent survey (Fortunato 2010). Optimization based methods (Di Martino and Sessa 2013; Mokryani et al. 2013; Srivastava et al. 2014) are a main class of methods to detect communities in complex networks (Cai et al. 2014c). Modularity, presented in (Girvan and Newman 2002), is the most used and best known quality function for optimization based methods to detect communities. Newman (2004) presented an agglomerative hierarchical method that searches for optimal values of modularity. Clauset et al. (2004) presented a fast implementation of the agglomerative hierarchical method. Blondel et al. (2008) presented a variant of the hierarchical agglomerative clustering approach, named BGLL. BGLL includes two phases that partitions a large network by repeatedly optimizing the modularity. Chaturvedi et al. (2012) applied BGLL to various large networks. Pizzuti (2008) presented a genetic algorithm for community detection in social networks, named GA-Net. Li and Song (2013) presented an extended compact genetic algorithm for community detection. Cai et al. (2014c, 2015) presented a discrete particle swarm optimization algorithm and a clonal selection algorithm for community detection. All of the above works consider community detection as a single objective optimization problem.

Communicated by V. Loia.

✉ Jiahai Wang
wjiahai@hotmail.com

Yalan Zhou
zhouylan@163.com

¹ College of Information, Guangdong University of Finance and Economics, Guangzhou, People's Republic of China

² Department of Computer Science, Sun Yat-sen University, Guangzhou, People's Republic of China

³ School of Mobile Information Engineering, Sun Yat-sen University, Guangzhou, People's Republic of China

A network may have multiple potential community structures in the real world; hence a more suitable approach is to simultaneously consider several conflicting objectives in community detection to discover complex and comprehensive community structures. Recently, several multiobjective optimization algorithms for community detection in networks have been proposed. Pizzuti (2012) presented a multiobjective genetic algorithm, named MOGA-Net. Shi et al. (2012) presented a multiobjective community detection algorithm including two phases, named MOCD. Gong et al. (2012a, 2014) presented a multiobjective evolutionary algorithm with decomposition, named MOEA/D-Net, and a multiobjective discrete particle swarm optimization algorithm based on decomposition, named MODPSO.

Most existing multi-objective community detection algorithms are based on evolutionary algorithms. In this study, we propose a local search-based multiobjective optimization algorithm, named MOLS-Net, to discover communities in networks. In the proposed algorithm, different objective-wise local searches are designed for different objectives. Extensive experiments on both synthetic and real-world networks show that MOLS-Net obtains better or competitive results compared with MODPSO, MOEA/D-Net, GA-Net, and BGLL.

The remaining sections are organized as follows. Section 2 gives the problem formulation. Section 3 proposes MOLS-Net for community detection. Section 4 presents experimental results. Finally, Sect. 5 presents conclusions.

2 Problem formulation

2.1 Formulation

Given an undirected network denoted as $G = (V, E)$, where V ($|V| = n$) and E are the sets of vertices and edges, respectively, G can be represented as an adjacency matrix A . The size of A is $|V| \times |V|$. $A_{ij} = 1$, if $e = (i, j) \in E$; otherwise, $A_{ij} = 0$. The community structure of the network can be regarded as a partition $P = (V_1, V_2, \dots, V_i, \dots, V_m)$ with the number m of V , where $V_i \subseteq V$, $V_i \neq \emptyset$, $\bigcup_{i=1}^m V_i = V$ and $V_i \cap V_j = \emptyset$ ($i \neq j$) should be satisfied for each V_i ($i = 1, 2, \dots, m$). V_i corresponds to a community of the network. Vertex–vertex connections are dense in a community V_i and are relatively sparse between V_i and V_j ($j = 1, 2, \dots, m$ and $j \neq i$). A community can be classified into a strong community and a weak community in a more formal definition (Radicchi et al. 2004).

In this study, ratio cut (RC) (Wei and Cheng 1991) and kernel K -means (KKM) (Gong et al. 2014) are used as the two objective functions to be minimized. RC, denoted as f_1 , measures the sum of the external link density between communities. KKM, denoted as f_2 , measures the sum of the internal link density in the same community. The two objectives f_1 and f_2 are formalized as

$$f_1 = \text{RC} = \sum_{i=1}^m \frac{L(V_i, \bar{V}_i)}{|V_i|}, \quad (1)$$

$$f_2 = \text{KKM} = 2(n - m) - \sum_{i=1}^m \frac{L(V_i, V_i)}{|V_i|}, \quad (2)$$

where n is the number of vertices in a network, m is the number of communities in a partition (solution), $|V_i|$ is the number of vertices in community i , $V_i \in P$, $\bar{V}_i = P - V_i$, $L(V_i, \bar{V}_i) = \sum_{i \in V_i, j \in \bar{V}_i} A_{ij}$, and $L(V_i, V_i) = \sum_{i, j \in V_i} A_{ij}$.

Minimizing f_1 and f_2 can ensure sparse interconnections and dense intraconnections. That is, optimizing RC tends to divide a network into large communities, while optimizing KKM tends to divide a network into small communities. This is in accordance with the characteristics of the communities in a network (Gong et al. 2014).

2.2 Multiobjective optimization

A multiobjective optimization problem (MOP) can be described as follows:

$$\min F(X) = (f_1(X), f_2(X), \dots, f_k(X)), \quad (3)$$

where $X = (X_1, X_2, \dots, X_D) \subseteq \mathfrak{R}^D$ is a vector in the D -dimensional solution space; $F \in \Omega^k$ is the objective space with k minimization objectives.

Given two feasible solutions X and Y , we say that X dominates Y if $\forall i: f_i(X) \leq f_i(Y)$ and $\exists j: f_j(X) < f_j(Y)$. Moreover, X is said to be Pareto optimal if it is not dominated by any other feasible solution. Then, the aim of multiobjective optimization is to find the set of Pareto optimal solutions, usually called Pareto set. The set of all Pareto optimal objective vectors is called Pareto front.

Over the years, a number of metaheuristics have been extended to solve MOPs. Multiobjective evolutionary algorithms (MOEAs) strive to obtain an accurate and well-distributed approximation of the true Pareto front. Popular MOEAs include nondominated sorting genetic algorithm II (NSGA-II) (Deb et al. 2002) and MOEA based on decomposition (MOEA/D) (Zhang et al. 2007; Rubio-Largo et al. 2014). More studies about the development and application of MOEAs can be found in a recent survey paper (Zhou et al. 2011). Besides MOEAs, local search-based algorithms, such as Pareto local search (Talbi et al. 2012) and multi-directional local search (Tricoire 2012), are promising alternative approaches to solve MOPs. The merit of local search-based algorithms is that problem-specific knowledge can be directly used to guide the search toward the Pareto front. Thus, they are specially suitable for multiobjective combinatorial optimization problems. More details about local search-based algorithms can be found in another recent survey paper (Talbi et al. 2012).

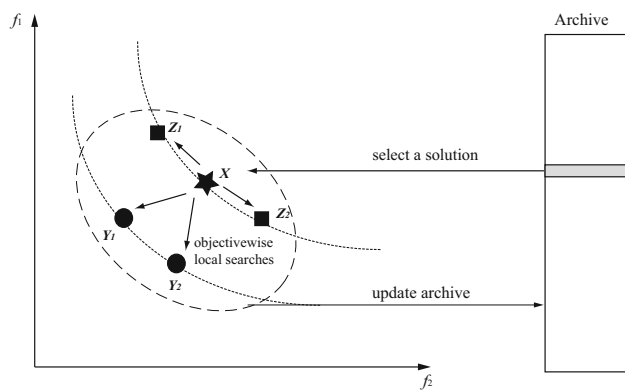


Fig. 1 Framework of MOLS-Net

In this study, a multiobjective local search (MOLS-Net) is proposed for community detection. Note that, in response to the particularities of a MOP, different multiobjective optimization algorithms may differ in the encoding scheme (responsible for the characterization of the search space), objective function, and operators that depend on the kind of encoding scheme adopted. As a consequence, MOLS-Net is different from the previous studies (Tricoire 2012; Zhou and Wang 2014; Wang et al. 2015) since MOLS-Net is composed of dedicated modules (for example, local search operators) to solve community detection problems.

3 Multiobjective local search for community detection

3.1 Framework of MOLS-Net

Figure 1 presents the framework of MOLS-Net. The feature of MOLS-Net is to use objectivewise local searches (Tricoire 2012; Zhou and Wang 2014; Wang et al. 2014, 2015). That is, for a given solution X , a local search $LS_1(X)$ is designed to improve the objective f_1 , while another local search $LS_2(X)$ is designed to improve the objective f_2 . An archive is maintained to keep all nondominated solutions found during the search. The main loop of MOLS-Net consists of (1) selecting a solution from archive, (2) performing objectivewise local searches to improve this solution, and (3) updating archive. The pseudo code of MOLS-Net is described in Algorithm 1.

In the following sections, solution representation, initialization, objectivewise local searches, and archive updating are described in detail.

3.2 Representation

A solution X in MOLS-Net is defined as $X = (x_1, x_2, \dots, x_i, \dots, x_n)$. Here, n is the number of the vertices in a network, and x_i is the integer community identifier of vertex v_i . The identifier can be any integer number between 1 and n . The

vertices having the same community identifier are considered to be the same community. A network with n vertices can be divided into n communities at most. Figure 2 gives a simple example of how the solution is encoded and decoded. In Fig. 2, the network with seven vertices is divided into two communities.

This representation does not need know the exact number m of communities in advance. The number is automatically determined during the multiobjective optimization process. This representation is intuitive and easy to decode. It takes little computational complexity.

3.3 Initialization

An initialization population with good quality can speed up the convergence and promote diversity. In this study, a simple and fast initialization mechanism based on label propagation (Raghavan et al. 2007) is used. In this procedure (Raghavan et al. 2007), each vertex is marked by the given unique label. Then, all vertices are swept over in random sequential order at each iteration, and each vertex's label is decided according to the majority of its neighbors' labels during the process. The process converges when each vertex has the majority label of its neighbors. Finally, a group of vertices with identical labels after convergence is considered as a community.

3.4 Objectivewise local searches

Given a solution X , $LS_1(X)$ and $LS_2(X)$ are designed to optimize objective f_1 and f_2 , respectively. The procedures are described in Algorithms 2 and 3, respectively.

In Algorithms 2 and 3, the neighbors of a community V_i are the communities that have at least one edge connection with the community V_i .

The merging operator in Algorithm 2 decreases the number of communities, which thus decreases the value of objective f_1 . The dividing operator in Algorithm 3 increases the number of communities, which thus decreases the value of objective f_2 (Angelimi et al. 2007). In this way, f_1 and f_2 will be minimized by the merging and the dividing operators, respectively.

In Algorithm 3, the idea of BGLL in Blondel et al. (2008) is used. The method consists of two phases. In the first phase, each vertex is considered as a community. The new communities are formed by finding the modularity change. The change is made by moving a vertex into the community of its neighborhood vertex. A vertex is placed into the community when the modularity gain of the community is maximum. This process is applied to all vertices until no vertex moving can improve the value of modularity. In the second phase, a new network is built by considering the communities found in the first phase as the new vertices. The weight of the link between the two vertices is the sum of the weights of the

Algorithm 1 Pseudo code of MOLS-Net

1: **parameters:** *init_size*: the number of initial solutions, *t*: running time, *step_parameter*: controlling search step.
input: the adjacent matrix *A* of a network.
output: archive.

2: **initialization:** initialize population with *init_size* size, generate several nondominated solutions to initialize *archive*.

3: **while** $t \leq$ maximum computation time **do**

4: randomly select a solution *X* from *archive*.

5: get the community number *m* of *X*.

6: generate a pseudo random number *q*, $q \in [1, \max(m * \text{step_parameter}, 3)]$.

7: **for** $i = 1, 2, \dots, q$, **do**

8: perform objectivewise local search $LS_1(X)$ and update *archive*.

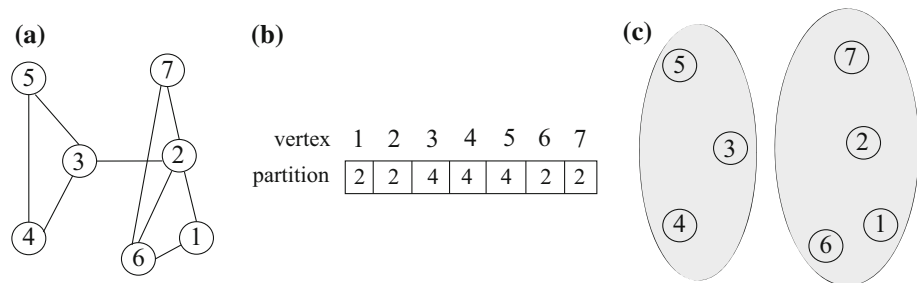
9: perform objectivewise local search $LS_2(X)$ and update *archive*.

10: **end for**

11: **end while**

12: return the *archive*.

Fig. 2 An example of solution representation. **a** graph topology, **b** solution representation, **c** community structure

**Algorithm 2** Pseudo code of local search $LS_1(X)$ for f_1

1: randomly select two communities from the solution *X*.

2: select the community with the larger external link density.

3: **if** $\text{rand}(0, 1) < 0.5$

4: **merge** the selected community with a community randomly selected from its neighbors.

5: **else**

6: change the label identifiers of all vertices in the selected community to the label identifier possessed by the majority of its neighbor vertices.

7: **end if**

Algorithm 3 Pseudo code of local search $LS_2(X)$ for f_2

1: randomly select two communities from the solution *X*.

2: select the community with the smaller internal link density.

3: **if** $\text{rand}(0, 1) < 0.5$

4: **divide** the selected community into two new communities by BGLL.

5: **else**

6: change the label identifiers of all vertices in the selected community to the label identifier possessed by the majority of its neighbor vertices.

7: **end if**

links in the corresponding communities. The method in the first phase is repeated for the new network. The method stops until the community number is smaller than or equal to 2. The algorithm returns the communities found.

During the objectivewise local search process as shown in Fig. 1, two situations may occur: (1) the newly generated solution, for example, Y_1 or Y_2 , dominates the original solution *X*; or (2) the newly generated solution, for example, Z_1 or Z_2 , is noncomparable with the original solution *X*. The first case occurs when the local search improves the corresponding objective without deteriorating other objectives. This means that a new dominating solution is found. This case deals with the convergence issue in MOPs since it

drives the algorithm to convergence toward Pareto front. The second case occurs when the local search improves the corresponding objective but deteriorating other objectives. This case deals with the diversity issue in MOPs since it drives the algorithm to spread along Pareto front. Both situations occur during the whole optimization process of local search. Therefore, it can achieve both the convergence and diversity goals of multiobjective optimization.

3.5 Archive updating

An archive is maintained to store the nondominated solutions found during the search process. Once a new solution

is generated by $LS_1(X)$ or $LS_2(X)$, the archive is updated to avoid missing any nondominated solution during the search process.

3.6 Complexity

The complexity of $LS_1(X)$ is $O(l)$, and the complexity of $LS_2(X)$ is $O(l \cdot n^2)$. Here, n and l denote the number of vertices and edges of a network, respectively. The whole complexity of MOLS-Net is $O(l \cdot n^3)$. The main time complexity of MOLS-Net lies in the divide operator in $LS_2(X)$. In the divide operator, the idea of BGLL (Blondel et al. 2008) is used. Experiment results in Blondel et al. (2008) and Chaturvedi et al. (2012) show that BGLL is extremely fast in practice. Hence, MOLS-Net in our experiments is also very fast. The computation time of MOLS-net will be given later.

4 Experimental results

All experiments are implemented in C++ on a PC [Intel(R) Core(TM) i5-4440 CPU @ 3.10 GHz processor with 8.0G RAM on 64 bits windows 7].

MOLS-Net is compared with two state-of-the-art MOEAs (i.e., MODPSO and MOEA/D-Net), two famous single objective optimization algorithms (i.e., GA-Net and BGLL). The number of generations of EA-based algorithms are all set as 400. Other parameters in these compared algorithms are set according to their corresponding references:

- (1) MODPSO (Gong et al. 2014): the population size is 100; the neighborhood parameter is 40; and the mutation rate is 0.1.
- (2) MOEA/D-Net (Gong et al. 2012a): the population size is 100; the neighborhood parameter is 10; the cross over rate is 0.9; the mutation rate is 0.06; and the update size is 2.
- (3) GA-Net (Pizzuti 2008): the population size is 100; the crossover rate is 0.8; and the mutation rate is 0.2.

The source codes of these algorithms were reimplemented in our experiments for fair comparison. All algorithms were run 30 times independently. The parameters of MOLS-Net are set as follows: the initial population size is $init_size = 100$; the parameter of controlling search step is $step_parameter = 0.3$; the maximum computation time is set as the average running time of MODPSO over 30 runs for fair comparison.

4.1 Evaluation metrics

Normalized mutual information (NMI) and modularity (Q) are two popular evaluation metrics to evaluate the quality of

the partition obtained. NMI is an external measure to estimate the similarity between the true partitions and the detected ones (Wu and Huberman 2004). Q is an internal measure to evaluate the extent to which the detected community structure deviates from randomness (Girvan and Newman 2002).

Let R denote a real partition of a network. Let P denote a partition found by an algorithm. NMI(R, P) is then defined as

$$NMI(R, P) = \frac{-2 \sum_{i=1}^{M_R} \sum_{j=1}^{M_P} M_{ij} \log \frac{M_{ij} n}{M_{i.} M_{.j}}}{\sum_{i=1}^{M_R} M_{i.} \log \frac{M_{i.}}{n} + \sum_{j=1}^{M_P} M_{.j} \log \frac{M_{.j}}{n}}, \quad (4)$$

where M_R and M_P , respectively, denote the number of real communities and found communities; M denotes a confusion matrix whose element M_{ij} is the number of vertices sharing in common by community i in R and community j in P ; $M_{i.}$ and $M_{.j}$, respectively, denote the sum over row i and column j of matrix M ; and n is the number of vertices of the network.

$NMI(R, P) = 1$ if $R = P$; $NMI(R, P) = 0$ if R and P are totally different. Obviously, the larger the value of NMI, the better the partition obtained.

Q is defined as

$$Q = \frac{1}{2l} \sum_{i,j} (A_{ij} - d_i d_j / 2l) \delta(V_i, V_j), \quad (5)$$

where the sum runs over all pairs of vertices of a network; l is the total number of edges; A_{ij} is the element of the adjacency matrix A of graph; d_i is the degree (the number of links that have connections with vertex i) of vertex i ; d_j is the degree of vertex j ; and $\delta(V_i, V_j) = 1$ if vertex i and j belong to the same community (i.e., $V_i = V_j$), otherwise $\delta(V_i, V_j) = 0$.

The larger the value of Q , the stronger the community structure.

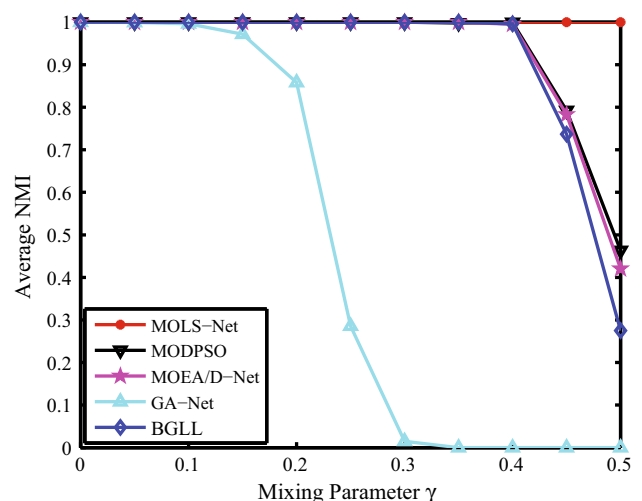


Fig. 3 Average NMI values obtained by the five algorithms on GN

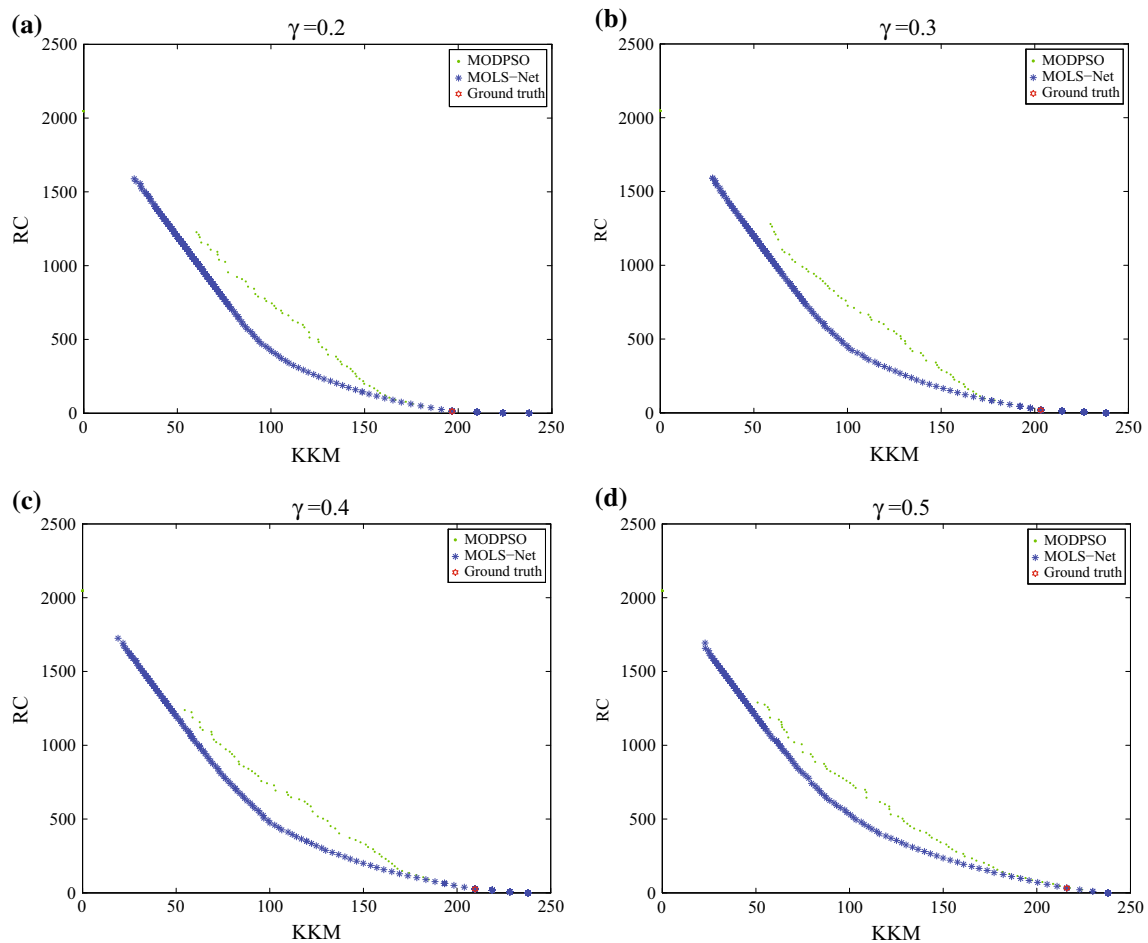


Fig. 4 Plots of nondominated solutions found by MOLS-Net and MODPSO on GN extended benchmark networks. **a** $\gamma = 0.2$, **b** $\gamma = 0.3$, **c** $\gamma = 0.4$, **d** $\gamma = 0.5$

4.2 Experiments on GN extended benchmark networks

GN extended benchmark network (Lancichinetti et al. 2008) is an extension of the classical benchmark proposed in Girvan and Newman (2002). The network has 128 vertices partitioned by four communities of 32 vertices each. Every vertex has an average degree of 16. Each vertex shares a fraction γ of links with the vertices of its community, and $1 - \gamma$ with the vertices of the other communities. γ is called the mixing parameter. When $\gamma < 0.5$, the neighbors of a vertex inside its community are more than the neighbors belonging to the other three communities. Eleven different networks with the value of γ increasing from 0 to 0.5 with an interval of 0.05 were generated in our experiments. The community structure of the network gradually becomes fuzzy as the value of γ increases.

MOLS-Net, MODPSO, and MOEA/D-Net generate a set of solutions on each run. The solution with maximum NMI value over each run is chosen as the output of each algorithm. The average values of NMI over 30 runs for the five algorithms are calculated and shown in Fig. 3. When the mix

parameter γ is smaller than 0.4, the average NMI values of all the algorithms, except for GA-Net, equal to 1. When the mix parameter γ value ranges from 0.4 to 0.5, MOLS-Net can

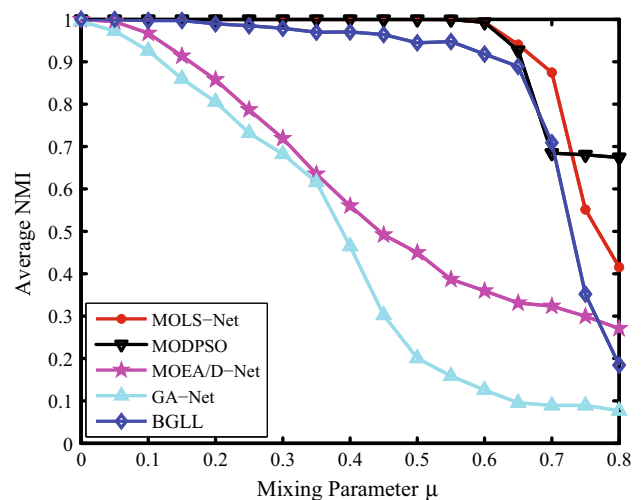


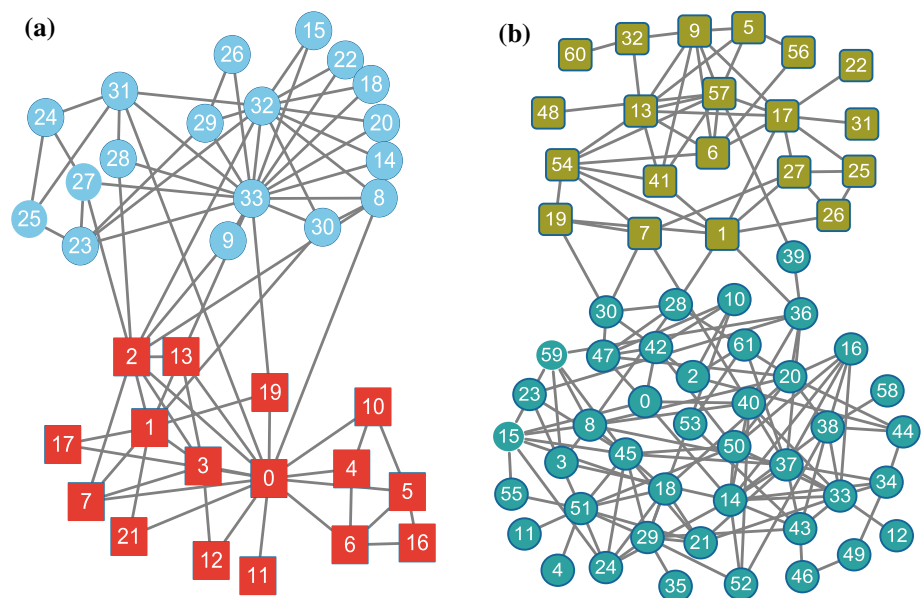
Fig. 5 Average NMI values obtained by the five algorithms on LFR

Table 1 Experimental results obtained by the five algorithms for real-world datasets

Network	Index	MOLS-Net	MODPSO	MOEA/D-Net	GA-Net	BGLL
Karate	NMI_{max}	1.0000	1.0000	1.0000	0.8255	0.7071
	NMI_{avg}	1.0000	0.9729	1.0000	0.6855	0.6280
	Q_{max}	0.4198	0.4198	0.3715	–	0.4198
	Q_{avg}	0.4198	0.4138	0.3715	–	0.4118
Dolphin	NMI_{max}	1.0000	1.0000	1.0000	0.4710	0.6363
	NMI_{avg}	1.0000	1.0000	1.0000	0.4176	0.5148
	Q_{max}	0.5268	0.5268	0.3735	–	0.5277
	Q_{avg}	0.5237	0.5196	0.3735	–	0.5210
Football	NMI_{max}	0.9361	0.9289	0.9367	0.9058	0.8923
	NMI_{avg}	0.9273	0.9245	0.9334	0.8691	0.8765
	Q_{max}	0.6046	0.6032	0.6005	–	0.6046
	Q_{avg}	0.6044	0.5995	0.5993	–	0.6038

The best results are marked in bold

Fig. 6 Real structures found by MOLS-Net. **a** on Zachary’s karate club network, **b** on dolphin social network



still successfully detect the real community structure (i.e., the average NMI values are 1). With MODPSO, MOEA/D-Net, and BGLL, the true structure become harder and harder to figure out.

Additionally, nondominated solutions found by MOLS-Net (denoted as $*$) and MODPSO (denoted as \bullet) from all runs on the selected GN networks for $\gamma = 0.2, \gamma = 0.3, \gamma = 0.4, \gamma = 0.5$ are shown in Fig. 4. The ground truths of these networks are known, and thus they are also presented in the figure. It is clear that MOLS-Net finds better nondominated solutions in terms of both convergence and diversity on the selected networks.

4.3 Experiments on LFR benchmark networks

GN extended benchmark networks have two drawbacks: (1) each vertex has the same degree, and (2) each community

has the same size. Hence, [Lancichinetti et al. \(2008\)](#) proposed a new benchmark network, named LFR. In LFR, both the degrees of vertices and the sizes of communities obey exponential distribution. Each vertex shares a fraction μ of links with the vertices of its community and a fraction $1 - \mu$ with the vertices of the other communities. μ is the mixing parameter. Seventeen different networks with the value of μ increasing from 0 to 0.8 with interval 0.05 were generated in our experiments.

Figure 5 displays the average NMI values obtained by the five algorithms on LFR benchmark networks. It can be seen that only MOLS-Net and MODPSO can successfully figure out the true community structure when the mix parameter μ value ranges from 0.15 to 0.6. All the algorithms cannot find true partition when μ is larger than 0.6. MOLS-Net outperforms MODPSO with μ ranging from 0.6 to 0.7, while

MOLS-Net is outperformed by MODPSO in the remaining cases.

4.4 Experiments on real-world benchmark networks

In order to further compare the performance of different algorithms, we use three real social networks which are available in Newman (2013). These networks are Zachary's karate club network, dolphin social network, and American college football network. These networks are widely used as benchmarks in community detection (Gong et al. 2012a, 2014; Shi et al. 2012; Pizzuti 2012).

Zachary's karate club network was constructed by Zachary. Zachary observed 34 members of a karate club over a period of two years. During this period, a disagreement developed between the administrator and instructor in the club. Thus, the instructor left and started a new club. The network split nat-

urally into two communities. We test a simple un-weighted version of Zachary's network.

Dolphin social network represents a network of 62 bottlenose dolphins. A tie between two dolphins was established by their statistically significant frequent association. The network naturally is divided into two large groups: the female group and the male one.

American college football network comes from United States college football. The network represents the schedule of Division I games during the 2000 season. Vertices in the graph represent teams, and edges represent regular season games between the two teams they connect. The teams are divided into conferences. The teams on average played 4 inter-conference matches and 7 intra-conference matches. The teams tended to play between members of the same conference. The network consists of 115 vertices and 616 edges grouped into 12 teams.

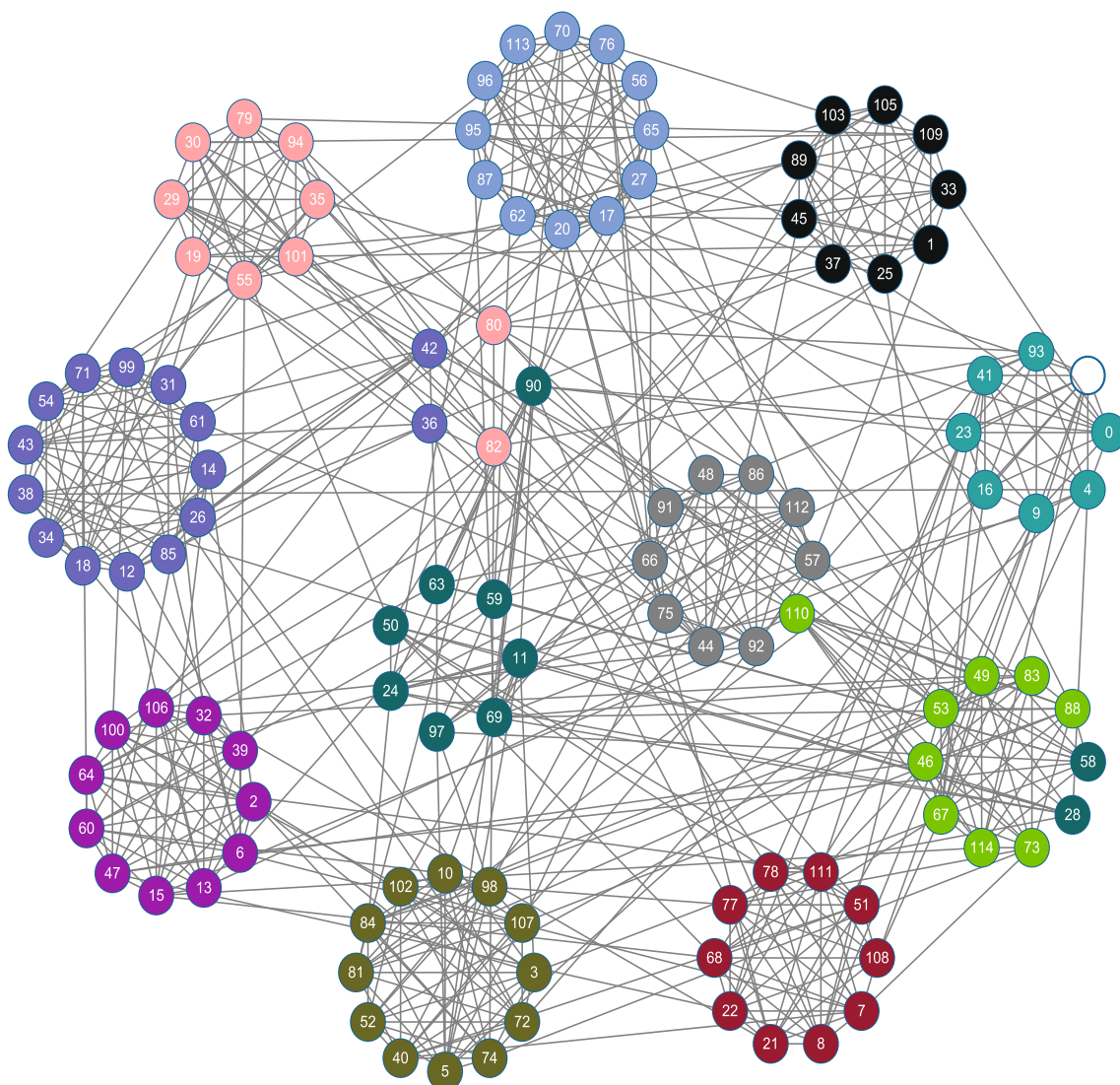


Fig. 7 The detected structure with $NMI = 0.9361$ on football network by MOLS-Net

Table 2 The computation time (seconds) of MOLS-Net on the three kinds of networks

GN Extended	$\gamma = 0.0$	$\gamma = 0.05$	$\gamma = 0.1$	$\gamma = 0.15$	$\gamma = 0.2$	$\gamma = 0.25$	$\gamma = 0.3$	$\gamma = 0.35$	$\gamma = 0.4$	$\gamma = 0.45$	$\gamma = 0.5$
Time	4.04	3.67	3.91	4.05	4.04	4.07	4.03	3.91	4.00	3.93	3.91
LFR	$\mu = 0.0$	$\mu = 0.05$	$\mu = 0.1$	$\mu = 0.15$	$\mu = 0.2$	$\mu = 0.25$	$\mu = 0.3$	$\mu = 0.35$	$\mu = 0.4$	$\mu = 0.45$	$\mu = 0.5$
Time	36.71	37.84	41.30	46.03	46.96	50.05	50.10	50.98	51.19	52.43	52.48
LFR	$\mu = 0.55$	$\mu = 0.6$	$\mu = 0.65$	$\mu = 0.7$	$\mu = 0.75$	$\mu = 0.8$					
Time	47.53	46.75	48.71	51.30	50.01	52.00					
Real-world	Karate	Dolphin	Football								
Time	0.66	1.06	2.45								

Evaluation metric Q has a resolution limit (Fortunato and Barthelemy 2007). A higher modularity often does not correspond to a better network partition (Fortunato 2010). Thus, in our experiments, the solution with the maximum NMI value is selected and the corresponding Q value is computed for MOLS-Net, MODPSO and MOEA/D-Net at each run. The maximum and average values of NMI (NMI_{\max} and NMI_{avg}) and Q (Q_{\max} and Q_{avg}) over all the five algorithms are reported in Table 1. Only NMI values are available in GA-Net. Table 1 indicates that MOLS-Net outperforms all the compared algorithms on Zachary's karate club network and dolphin social network. MOLS-Net can always detect the real community structure on those two networks (correspond to $NMI_{\max} = 1$ and $NMI_{\text{avg}} = 1$). Furthermore, the largest Q values are obtained. The detected real communities on Zachary's karate club network and dolphin social network by MOLS-Net are presented in Fig. 6a, b, respectively. Different colors of the vertices indicate the different communities obtained by MOLS-Net.

On American college football network, from the perspective of Q , MOLS-Net outperforms all the comparison algorithms. From the perspective of NMI, MOLS-Net is outperformed by MOEA/D-Net. Due to the self-complicated structure of the network, none of the algorithms can find the true partition. The detected communities with $NMI = 0.9361$ of MOLS-Net on this complex network are presented in Fig. 7. From Fig. 7, it is clear that some vertices have been misplaced.

4.5 Computation time of MOLS-Net

The computation times of MOLS-Net (i.e., the average running time of MODPSO) on the three kinds of networks are presented in Table 2. From Table 2, we can conclude that the computation time of MOLS-Net is reasonable.

5 Conclusion

In this study, a new multiobjective community detection algorithm (MOLS-Net) has been proposed. MOLS-Net simulta-

neously optimizes two conflicting objective functions, RC and KKM. In MOLS-Net, each objective is optimized by the corresponding objectivewise local search. The experimental results show that MOLS-Net performs better than or competitive with the existing algorithms including MODPSO, MOEA/D-Net, GA-Net, and BGLL on synthetic and real-world networks.

In the future, MOLS-Net will be extended to detect community in signed social networks (Liu et al. 2014; Li et al. 2014; Cai et al. 2014a), dynamic social networks (Gong et al. 2012b), and overlapping communities in social networks (Xie et al. 2013). In addition, more powerful objectivewise local searches for the objectives should be further developed to improve performance.

Acknowledgments This work was supported in part by Zhujiang New Star Program of Science and Technology in Guangzhou (2012J2200085), Excellent Young Teachers Training Program in Guangdong Colleges and Universities (Yqgdufe1404), Program for Characteristic Innovation Talents of Guangdong (2014KTSCX127), and the National Natural Science Foundation of China (61472453, U1401256).

Conflict of interest The authors declare that they have no conflict of interest. This research does not involve any human participant or animal and thus has no informed consent.

References

- Angelini L, Boccaletti S, Marinazzo D, Pellicoro M, Stramaglia S (2007) Identification of network modules by optimization of ratio association. *Chaos* 17(2):023114
- Blondel V, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *J Stat Mech: Theory Exp* 2008(10):P10008
- Cai Q, Gong M, Shen B, Ma L, Jiao L (2014a) Discrete particle swarm optimization for identifying community structures in signed social networks. *Neural Networks* 58:4–13
- Cai Q, Gong M, Ma L, Ruan S, Yuan F, Jiao L (2015) Greedy discrete particle swarm optimization for large-scale social network clustering. *Inf Sci*. doi:10.1016/j.ins.2014.09.041
- Cai Q, Gong M, Ma L, Jiao L (2014b) A novel clonal selection algorithm for community detection in complex networks. *Comput Intell*. doi:10.1111/coin.12031 (in Press)
- Cai Q, Ma L, Gong M, Tian D (2014c) A survey on network community detection based on evolutionary computation. *Int J Bio-Inspired Comput* (in Press)

- Carullo G, Castiglione A, De Santis A, Palmieri F (2015) A triadic closure and homophily-based recommendation system for online social networks. *World Wide Web*. doi:10.1007/s11280-015-0333-5 (in Press)
- Castiglione A, Pizzolante R, De Santis A, Carpentieri B, Castiglione A, Palmieri F (2015) Cloud-based adaptive compression and secure management services for 3d healthcare data. *Future Generation Comput Syst* 43–44:120–134 (In Press)
- Chaturvedi P, Dhara M, Arora D (2012) community detection in complex network via BGLL algorithm. *Int J Comput Appl* 48(1):32–42
- Clauset A, Newman M, Moore C (2004) Finding community structure in very large networks. *Phys Rev E-Stat Nonlinear Soft Matter Phys* 70(2):066111
- 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
- Di Martino F, Sessa S (2013) A fuzzy particle swarm optimization algorithm and its application to hotspot events in spatial analysis. *J Ambient Intell Humaniz Comput* 4(1):85–97
- Esposito C, Ficco M, Palmieri F, Castiglione A (2013) Interconnecting federated clouds by using publish-subscribe service. *Clust Comput* 16(4):887–903
- Fortunato S, Barthelemy M (2007) Resolution limit in community detection. *Proc Natl Acad Sci USA* 104(1):36–41
- Fortunato S (2010) Community detection in graphs. *Phys Reports* 486(3–5):75–174
- Girvan M, Newman M (2002) Community structure in social and biological networks. *Proc Natl Acad Sci USA* 99(12):7821–7826
- Gong M, Ma L, Zhang Q, Jiao L (2012a) Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Physica A: Stat. Mech Appl* 391(15):4050–4060
- Gong MG, Zhang LJ, Ma JJ, Jiao LC (2012b) Community detection in dynamic social networks based on multiobjective immune algorithm. *J Comput Sci Technol* 27(3):455–467
- 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
- Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys Rev E Stat Nonlinear Soft Matter Phys* 78(4):046110
- Li J, Song Y (2013) Community detection in complex networks using extended compact genetic algorithm. *Soft Comput* 17(6):925–937
- Li Y, Liu J, Liu C (2014) A comparative analysis of evolutionary and memetic algorithms for community detection from signed social networks. *Soft Comput* 18(2):329–348
- Liu C, Liu J, Jiang Z (2014) A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. *IEEE Trans Cybern* 44(12):2274–2287
- Li J, Wang Q, Wang C, Cao N, Ren K, Lou W (2010) Fuzzy keyword search over encrypted data in cloud computing. In: *Proceedings of the 29th IEEE International Conference on Computer Communications*, pp 441–445
- Mokryani G, Siano P, Piccolo A (2013) Optimal allocation of wind turbines in microgrids by using genetic algorithm. *J Ambient Intell Humaniz Comput* 4(6):613–619
- Newman M (2013) Network data sets. <http://www-personal.umich.edu/~mejn/netdata/>
- Newman M (2004) Fast algorithm for detecting community structure in networks. *Phys Rev E Stat Nonlinear Soft Matter Phys* 69(2):066133
- Pizzuti C (2008) GA-Net: a genetic algorithm for community detection in social networks. In: *Lecture notes in computer science*, vol 5199. LNCS, pp 1081–1090
- Pizzuti C (2012) A multiobjective genetic algorithm to find communities in complex networks. *IEEE Trans Evol Comput* 16(3):418–430
- Radicchi F, Castellano C, Cecconi F, Loreto V, Paris D (2004) Defining and identifying communities in networks. *Proc Natl Acad Sci USA* 101(9):2658–2663
- Raghavan U, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. *Phys Rev E Stat Nonlinear Soft Matter Phys* 76(3):036106
- Rubio-Largo A, Zhang Q, Vega-Rodriguez M (2014) Multiobjective evolutionary algorithm based on decomposition for 3-objective optimization problems with objectives in different scales. *Soft Comput* 19(1):157–166
- Shi C, Yan Z, Cai Y, Wu B (2012) Multi-objective community detection in complex networks. *Appl Soft Comput J* 12(2):850–859
- Srivastava V, Tripathi B, Pathak V (2014) Biometric recognition by hybridization of evolutionary fuzzy clustering with functional neural networks. *J Ambient Intell Humaniz Comput* 5(4):525–537
- Talbi EG, Basseur M, Nebro A, Alba E (2012) Multi-objective optimization using metaheuristics: non-standard algorithms. *Int Trans Oper Res* 19(1–2):283–305
- Tricoire F (2012) Multi-directional local search. *Comput Oper Res* 39(12):3089–3101
- Wang J, Zhong C, Zhou Y, Zhou Y (2014 In press) Multiobjective optimization algorithm with objective-wise learning for continuous multiobjective problems. *J Ambient Intell Humaniz Comput*. (In Press)
- Wang J, Zhou Y, Wang Y, Zhang J, Chen CP, Zheng Z (2015) Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances and algorithms. *IEEE Trans Cybern*. (In Press)
- Wei YC, Cheng CK (1991) Ratio cut partitioning for hierarchical designs. *IEEE Trans Comput-Aided Des Integr Circuits Syst* 10(7):911–921
- Wu F, Huberman B (2004) Finding communities in linear time: a physics approach. *Eur Phys J B* 38(2):331–338
- Xie J, Kelley S, Szymanski B (2013) Overlapping community detection in networks: the state-of-the-art and comparative study. *ACM Comput Surv* 45(4):1–36
- Zhang Q, Member S, Li H (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multi-objective evolutionary algorithms: a survey of the state-of-the-art. *Swarm Evol Comput* 1(1):23–49
- Zhou Y, Wang J (2014) A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows. *IEEE Syst J*. (In Press)