Gong MG, Zhang LJ, Ma JJ *et al.* Community detection in dynamic social networks based on multiobjective immune algorithm. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(3): 455–467 May 2012. DOI 10.1007/s11390-012-1235-y

Community Detection in Dynamic Social Networks Based on Multiobjective Immune Algorithm

Mao-Guo Gong (公茂果), *Senior Member*, *CCF*, *Member, ACM*, *IEEE*, Ling-Jun Zhang (张岭军), Jing-Jing Ma (马晶晶), and Li-Cheng Jiao (焦李成), *Senior Member*, *CCF*, *IEEE*

Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University Xi'an 710071, China

E-mail: gong@ieee.org; lingjun528@163.com; jjma@ieee.org; lchjiao@mail.xidian.edu.cn

Received August 29, 2011; revised December 13, 2011.

Abstract Community structure is one of the most important properties in social networks, and community detection has received an enormous amount of attention in recent years. In dynamic networks, the communities may evolve over time so that pose more challenging tasks than in static ones. Community detection in dynamic networks is a problem which can naturally be formulated with two contradictory objectives and consequently be solved by multiobjective optimization algorithms. In this paper, a novel multiobjective immune algorithm is proposed to solve the community detection problem in dynamic networks. It employs the framework of nondominated neighbor immune algorithm to simultaneously optimize the modularity and normalized mutual information, which quantitatively measure the quality of the community partitions and temporal cost, respectively. The problem-specific knowledge is incorporated in genetic operators and local search to improve the effectiveness and efficiency of our method. Experimental studies based on four synthetic datasets and two real-world social networks demonstrate that our algorithm can not only find community structure and capture community evolution more accurately but also be more steadily than the state-of-the-art algorithms.

Keywords community detection, community evolution, multiobjective optimization, evolutionary algorithm, social network

1 Introduction

In recent years, the research on social networks is becoming more and more important. Especially, their time-evolving version, dynamic networks are attracting increasing interest due to their great potential in capturing natural and social phenomena over time[1]. As an example, the evolution of informal groups within a large organization can provide insight into the organization's global decision-making behaviour.

Social networks are usually represented by graphs where nodes represent individuals and edges represent relationships and interactions among individuals. Based on this graph representation, there has been a large body of work on analyzing communities in static social networks, but only a few studies examined the dynamics of communities in evolving social networks. Previous studies usually adopt two-step approach where first static analysis is applied to the snapshots of the social network at different time steps, and then community evolution is introduced afterward to interpret the change of communities over time[2]. However, data from real-world networks are ambiguous and subject to noise. Under such scenarios, if an algorithm extracts community structure for each time step independently of other time steps, it often results in community structure with high temporal variation[3].

Some more recent studies attempted to unify the processes of community extraction and evolution extraction by using certain heuristics, such as regularizing temporal smoothness. This idea comes from a new kind of clustering concept called evolutionary clustering which has been proposed to capture the evolutionary process of clusters in temporal data[4]. This framework assumes that the structure of clusters significantly changing in a very short time is less desirable, and so it tries to smooth out each community over time. Several methods of finding communities and their evolutions in

Regular Paper

This work was supported by the National High Technology Research and Development 863 Program of China under Grant No. 2009AA12Z210, the Program for New Century Excellent Talents in University of China under Grant No. NCET-08-0811, the Program for New Scientific and Technological Star of Shaanxi Province of China under Grant No. 2010KJXX-03, and the Fundamental Research Funds for the Central Universities of China under Grant No. K50510020001.

dynamic networks using this idea have been proposed successively, which will be described in Section 2. These methods try to maximize cluster accuracy, with respect to incoming data of the current time step, and minimize cluster drift from one time step to the successive one. In order to optimize both these two competing objectives, an input parameter that controls the preference degree of a user towards either the snapshot cost or the temporal cost is needed. In [5], Folino *et al.* formulated the community detection in dynamic networks as a multi-objective optimization problem to avoid fixing the parameter in advance. They adopted a multiobjective genetic algorithm to optimize the two objectives Community Score (CS) and Normalized Mutual Information (NMI) simultaneously. Following this work, in this paper, we introduce a novel multiobjective immune algorithm with local search to solve the community detection problem in dynamic networks. Experimental studies on synthetic datasets and real-world datasets demonstrate the effectiveness of our algorithm. Compared to the state-of-the-art algorithms, our algorithm can discover the community structure and their evolutions more accurately.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 describes the formulation and the basic framework evolutionary clustering. Section 4 describes the proposed multiobjective community detection algorithm. Experimental studies are presented in Section 5. The concluding remarks are given in the last section.

2 Related Work

Detecting communities is becoming an important research topic in social network analysis, web community analysis, applied physics, computer vision, machine learning, etc. In recent years, many effective static network detection approaches have been proposed by researchers successively. Girvan and Newman proposed a divisive algorithm that uses edge betweenness as a metric to identify the boundaries of communities[6-7]. The algorithm is most popular and historically important. However, the algorithm makes heavy demands on computational resources, afterwards Newman proposed another fast algorithm based on the greedy optimization of the quantity known as modularity[8]. Later, Duch and Arenas proposed a new divisive algorithm that optimizes the modularity using a heuristic search based on the Extremal Optimization (EO) algorithm[9]. And other classic community detection algorithms based on modularity can be found in [10]. The first algorithm that finds both overlapping communities and the hierarchical structure was proposed by Lancichinetti *et al.*[11] Du *et al.* presented a faster algorithm ComTector which is more efficient for the community detection in large complex networks based on the nature of overlapping cliques[12].

Recently, finding communities and their evolutions in dynamic networks has gained more and more attention. Kumar *et al.* studied the evolution of the blogosphere as a graph in terms of the change of characteristics, the change of communities, as well as the burstiness in blog community[13]. Leskovec *et al.* studied the patterns of growth for graphs in various fields and proposed generators that produce graphs exhibiting the discovered patterns[14]. Palla *et al.* analyzed a co-authorship network and a mobile phone network, where both networks are dynamic, by using the clique percolation method (CPM)[15]. Asur *et al.* introduced a family of events on both communities and individuals to characterize evolution of communities[16], and so on.

There are some recent studies on evolutionary clustering that are closely related to our work. Chakrabarti *et al.* proposed the first evolutionary clustering method as the problem of clustering data coming at different time steps to produce a sequence of clusterings[4]. It should take care of two potentially conflicting criteria: the current clustering should reflect as accurately as possible the data coming during the current time step; at the same time, the clustering should not shift dramatically from one time step to the successive. This framework assumes that the abrupt change of clustering in a short time period is not desirable, thus it smooths out each community over time by incorporating temporal smoothness at each time step. Based on the idea of evolutionary clustering, Sarkar and Moore proposed a dynamic method that embeds nodes into latent spaces where the locations of the nodes at consecutive time steps are regularized so that dramatic change is unlikely[17]. Chi *et al.* proposed an evolutionary version of the spectral clustering algorithm. They used graph cut as a metric for measuring community structure and community evolution[18]. Lin *et al.* extended the graph-factorization clustering (GFC) and proposed the FacetNet algorithm[3] for analyzing dynamic communities. In their algorithm, an iterative algorithm is guaranteed to converge to (local) optimal solutions by the monotonic decrease of the cost function. Ahmed and Xing extended temporal dirichlet process mixture model for clustering problem for documents[19]. Tang *et al.* used joint matrix factorization method to discover the community evolution[20]. Kim and Han proposed a particle-and-density based evolutionary clustering method able to deal with a variable number of communities between different time steps. The method introduces the concept of nano-community and l-clique-by-clique (l-KK) to discover a variable number of communities that can evolve, form,

and dissolve[21].

In fact, the detection of community structure with temporal smoothness can be formulated as a multiobjective optimization problem. The first objective is the maximization of the community quality, which measures how well the community structure found represents the network at the current time. The second objective is the minimization of the temporal cost, which measures the distance between two community structures at consecutive time steps. Thus, Folino *et al.* proposed a dynamic multiobjective genetic algorithm (DYN-MOGA) to discover communities in dynamic networks by employing genetic algorithm. The two objectives to be optimized are formulated as Community Score and NMI[5]. Another work is by Kim who proposed adaptive integration of multiobjective evolutionary algorithms based on NSGA-II particularly for online social network clustering[22].

3 Formulation

3.1 Notation

The dynamic network G is defined as a sequence of networks $G_t(V_t, E_t)$, i.e., $G = \{G_1, G_2, \ldots, G_T\}$, where V_t is a set of objects, each $v_i \in V_t$ represents an individual and each edge $v_{ij} \in E_t$ denotes the presence of interactions between v_i and v_j. We use G_t in the graph to represent the snapshot of the network N_t at time t. Let $S_t = \{C_t^1, C_t^2, \ldots, C_t^k\}$ denote the community structure of the network N_t at time t where C_t^i denotes the i-th community at time t.

3.2 Evolutionary Clustering

In order to analyze communities and their evolutions in a unified process, we use the community structure at time $t-1$ to regularize the community structure at time t. This framework is first proposed by Chakrabarti *et al.* to cluster dynamic data[4]. At each time step a new clustering must be produced by simultaneously optimizing two conflicting criteria. The first is that the clustering should reflect as accurately as possible the data coming during the current time step. The second is that each clustering should not shift dramatically from one time step to the successive. In order to satisfy the second condition, the temporal smoothness is defined to smooth out each community over time. Thus the cost function consisting of two parts is defined as follows:

$$\text{Cost} = \alpha \times SC + (1-\alpha) \times TC, \tag{1}$$

where, the snapshot cost SC measures how well a community structure S_t represents the network at time t. The temporal cost TC measures how similar the community structure S_t is with the previous community structure S_{t-1}. The parameter α is set by the user to control the level of emphasis on each part of the total cost. When $\alpha = 1$, the framework returns the clustering without temporal smoothing. When $\alpha = 0$, however, it produces the same clustering results with the previous time step, i.e., $S_t = S_{t-1}$.

Because of the better efficiency of evolutionary clustering, several representative frameworks for dynamic community detection have adopted this concept, which demonstrates it is very effective for community identification in dynamic social networks. Therefore, the multiobjective community detection algorithm in dynamic social networks proposed in this paper also borrows this idea. However, we try to optimize both the snapshot cost and the temporal cost without the need to fix the control parameter α, which will be described in the following sections.

3.3 Multiobjective Optimization

Multiobjective optimization seeks to optimize a vector of functions,

$$\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^{\mathrm{T}} \tag{2}$$

subject to

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in \boldsymbol{\Omega},$$

where $\boldsymbol{x}$ is called the decision vector, and $\boldsymbol{\Omega}$ is the feasible region in decision space.

Considering a maximization problem for each objective, it is said that a decision vector $\boldsymbol{x}_A \in \boldsymbol{\Omega}$ dominates another vector $\boldsymbol{x}_B \in \boldsymbol{\Omega}$ (written as $\boldsymbol{x}_A \succ \boldsymbol{x}_B$) if and only if

$$\begin{aligned} &\forall i = 1, 2, \ldots, k \ f_i(\boldsymbol{x}_A) \geqslant f_i(\boldsymbol{x}_B) \wedge \\ &\exists j = 1, 2, \ldots, k \ f_j(\boldsymbol{x}_A) > f_j(\boldsymbol{x}_B). \end{aligned} \tag{3}$$

We say that a vector of decision variables $\boldsymbol{x}^* \in \boldsymbol{\Omega}$ is a Pareto-optimal solution or nondominated solution if there does not exist another $\boldsymbol{x} \in \boldsymbol{\Omega}$ such that $\boldsymbol{x} \succ \boldsymbol{x}^*$.

Then the Pareto-optimal set is defined as

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

So the Pareto-optimal set is the set of all Pareto-optimal solutions. The corresponding image of the Pareto-optimal set under the objective function space

$$\begin{aligned} PF^* \triangleq \{&\boldsymbol{F}(x^*) = \\ &(f_1(\boldsymbol{x}^*), f_2(\boldsymbol{x}^*), \ldots, f_k(\boldsymbol{x}^*))^{\mathrm{T}} | \boldsymbol{x}^* \in P^*\} \end{aligned} \tag{4}$$

is called the Pareto-optimal front. The aim of a multiobjective optimization algorithm is to find a set of Pareto-optimal solutions approximating the true Pareto-optimal front.

In the last few years, many efforts have been devoted to the application of evolutionary computation to the development of multiobjective optimization algorithms. So far, a variety of multiobjective optimization algorithms have been proposed[24-28]. We also proposed a multiobjective optimization algorithm, named Nondominated Neighbor Immune Algorithm (NNIA) in [29]. NNIA adopts a novel nondominated neighbor-based selection technique, an immune inspired operator, two heuristic search operators, and elitism. It turns out that NNIA is a very effective method for the multiobjective optimization problems by a mass of experiments. Because of its good performance, the proposed dynamic multiobjective community detection algorithm is based on NNIA, which will be described in the next section.

4 Proposed Community Detection Algorithm Based on NNIA

4.1 Objective Functions

An important issue in community detection is how to quantitatively measure the quality of the community partitions. A quantitative definition, network modularity, proposed by Grivan and Newman[7], has been proved to be an effective objective function to detect communities in recent studies. The modularity of a partition of a network can be written as

$$Q = \sum_{s=1}^{m} \Big(\frac{l_s}{L} - \Big(\frac{d_s}{2L} \Big)^2 \Big), \tag{5}$$

where the sum is over the m communities of the partition, l_s is the number of links inside the s-th community, L is the total number of links in the network, and d_s is the total degree of the nodes in the s-th community. The first term of the summand in (4) is the fraction of edges inside a community, the second term is the expected value of the fraction of edges that would be in the network if edges fall at random without regard to the community structure. If the number of within-community edges is no better than random, we will get $Q = 0$. While the value $Q = 1$, which is the maximum, indicates a strong community structure obtained[7]. As described in Subsection 3.2, the cost function is composed by the two competing objectives. The first objective is the snapshot cost which measures how well a community structure S_t represents the data at time t. And modularity which not only maximizes the number of connections inside one community but also minimizes the number of links between the communities is the right objective function that we need.

The second objective function is the temporal cost which measures how similar the community structure S_t is with the previous community structure S_{t-1}. Thus we use NMI which estimates the similarity between two communities as the second objective function to maximize. NMI is a similarity measure proved to be reliable by Danon *et al.*[30] Given two partitions A and B of a network in communities, let $\boldsymbol{C}$ be the confusion matrix whose element $\boldsymbol{C}_{ij}$ is the number of nodes of community i of the partition A that are also in the community j of the partition B. $\mathrm{NMI}(A, B)$ is defined as

$$\mathrm{NMI}(A,B) = \frac{-2\sum_{i=1}^{\boldsymbol{C}_A}\sum_{j=1}^{\boldsymbol{C}_B} \boldsymbol{C}_{ij}\log(\boldsymbol{C}_{ij}n/\boldsymbol{C}_{i.}\boldsymbol{C}_{.j})}{\sum_{i=1}^{\boldsymbol{C}_A} \boldsymbol{C}_{i.}\log(\boldsymbol{C}_{i.}/n) + \sum_{j=1}^{\boldsymbol{C}_B} \boldsymbol{C}_{.j}\log(\boldsymbol{C}_{.j}/n)}, \tag{6}$$

where $\boldsymbol{C}_A(\boldsymbol{C}_B)$ is the number of groups in the partitioning $A(B)$, $\boldsymbol{C}_{i.}(\boldsymbol{C}_{.j})$ is the sum of the elements of $\boldsymbol{C}$ in row i (column j), and n is the number of nodes. From (5), we can know that if $A = B$, $\mathrm{NMI}(A, B) = 1$ and if A and B are completely different, $\mathrm{NMI}(A, B) = 0$. In this study, these two objectives to be optimized should be maximized simultaneously.

4.2 Representation

The locus-based adjacency representation (LAR) proposed by Park and Song[31] is adopted in this study. In this graph-based representation, an arbitrary individual g in the population consists of n genes, in which each gene corresponds to a node in the network and n denotes the total number of nodes in this network. And each gene i can take an arbitrary allele value j in the range $\{1, 2, \ldots, n\}$, which means a link between nodes i and j existing in the corresponding graph G of individual g. This also means nodes i and j will be in the same community in the network. In the decoding step, it is necessary to identify all the components of the corresponding graph. The nodes belonging to the same component are assigned to the same community. A main advantage of this representation is that the number k of clusters is automatically determined by the number of components contained in an individual and determined by the decoding step[32]. In addition, the decoding process can be done in a linear time as shown in [33], which illustrates this encoding schema is very effective for community detection. The LAR representation and the corresponding encoded genotype are shown in Fig.1.

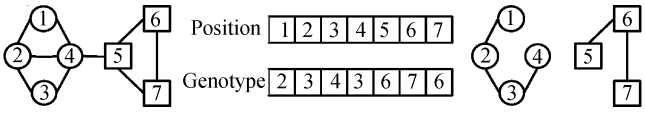

Fig.1. Illustration of the locus-based adjacency presentation.

4.3 Population Initialization

If an individual is randomly generated, some components in its corresponding graph G may be disconnected in the original network N, which also means G may be not a subgraph of N. Thus our initialization process takes in account the effective connections of the nodes in the network. For example, if an individual could contain an allele value j in the i-th position, where i must be one of neighbors of node i, i.e., there is a link between nodes i and j in its corresponding graph G. So we can guarantee the generated individuals are safe enough to avoid the meaningless divisions of the original network.

Some special operators, such as proportional cloning, uniform crossover and mutation used in our algorithm will be described in detail in the following subsections.

4.4 Main Loop of the Proposed Algorithm

The main loop of the dynamic multiobjective community detection algorithm based on NNIA with local search, termed as DYN-LSNNIA will be given in this subsection. In order to solve the problem of community detection in dynamic social networks, we should deal with the network at the initial time step firstly. Because there is no history information at time step 1, that is to say the temporal cost is zero, the network at time step 1 can be clustered without smoothing. So it needs to optimize only the first objective function, i.e., modularity, which is equivalent to the problem of single multiobjective optimization. As far as we know, GA-Net proposed by Pizzuti is an effective algorithm to discover communities in social networks by employing genetic algorithm[32]. Thus the new algorithm adopts GA-Net to process the initial network at time step 1, however, the objective function to be optimized is replaced by the modularity which is used in our algorithm. The main framework of the algorithm is as follows.

Algorithm 1. DYN-LSNNIA

Input: T (number of the time steps), $\{G_1, G_2, \ldots, G_T\}$ (sequence of dynamic network).

Output: $\{S_1, S_2, \ldots, S_T\}$ (sequence of community structure found in the dynamic network).

Step 1: Generate the initial clustering $S_1 = \{C_1^1, C_1^2, \ldots, C_1^k\}$ of the network G_1 with GA-Net. Set $ts = 1$.

Step 2: If $ts \geqslant T$ is satisfied, export the sequence of network $\{S_1, S_2, \ldots, S_T\}$ as the output, stop; otherwise, go to Step 3.

Step 3: Use the procedure of the revised NNIA adapted for community detection to process the network G_{ts} at time step ts. During this procedure, select the dominant population D_t in each generation.

Step 4: Perform the local search on the selected individuals in D_t to generate the new dominant population D_t'. Update the dominant population with D_t'. And then finish the other operations according the steps of the revised NNIA.

Step 5: Select the solution on the Pareto front, which has the maximum Community Score at the end of time step ts. Decode the selected individual to get the community structure $S_{ts} = \{C_{ts}^1, C_{ts}^2, \ldots, C_{ts}^k\}$ of the network G_{ts}.

Step 6: $ts = ts + 1$, and then return to Step 2.

4.5 Proportional Cloning

In this study, the proportional cloning T^C on the active population $A = \{a_1, a_2, \ldots, a_{|A|}\}$ is defined as

$$\begin{aligned} &T^C(a_1 + a_2 + \cdots + a_{|A|}) \\ &= T^C(a_1) + T^C(a_2) + \cdots + T^C(a_{|A|}) \\ &= \{a_1^1 + a_1^2 + \cdots + a_1^{q_1}\} + \{a_2^1 + a_2^2 + \cdots + a_2^{q_2}\} + \cdots + \\ &\quad \{a_{|A|}^1 + a_{|A|}^2 + \cdots + a_{|A|}^{q_{|A|}}\}, \end{aligned} \tag{7}$$

where

$$\begin{aligned} &T^C(a_i) = \{a_i^1 + a_i^2 + \cdots + a_i^{q_i}\}, \\ &a_i^j = a_i, \quad i = 1, 2, \ldots, |A|, \quad j = 1, 2, \ldots, q_i, \end{aligned}$$

q_i is a self-adaptive parameter. The representation $+$ is not the arithmetical operator, but only separates the antibodies here. $q_i = 1$ denotes that there is no cloning on antibody a_i. The individual with greater crowding-distance value is reproduced more times, therefore, the individual with greater crowding-distance value has a larger q_i. Because the crowding-distance values of boundary solutions are positive infinity, before computing the value of q_i for each active antibody, we set the crowding-distance value of the boundary individuals (in objective space) to be equal to the double value of the maximum value of active antibodies except the boundary individuals. Then the value of q_i is calculated as

$$q_i = \left\lceil n_C \times \frac{\zeta(a_i, A)}{\sum_{j=1}^{|A|} \zeta(a_j, A)} \right\rceil, \tag{8}$$

where $\zeta(a_j, A)$ denotes the crowding-distance value of the active antibodies a_j, n_C is an expectant value of the size of the clone population.

Fig.2 illustrates the procedure of proportional cloning. All the antibodies in subpopulation $\{a_i^1, a_i^2, \ldots, a_i^{q_i}\}$ are the result of the cloning on antibody a_i, and have the same property as a_i. In fact, cloning on antibody a_i is to make multiple identical copies of a_i. The aim is that the greater the crowding-distance value of an individual, the more times the individual will be

reproduced. So there exist more chances to search in less-crowded regions of the trade-off front.

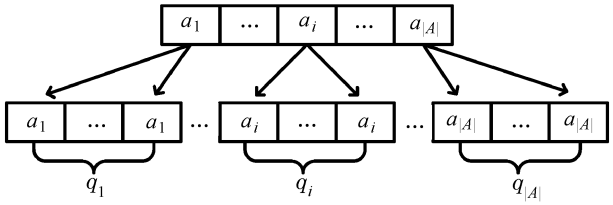

Fig.2. Illustration of the proportional cloning.

4.6 Uniform Crossover and Mutation

To guarantee the maintenance of the effective connections of the nodes in the social network in the child individual, we adopt the uniform crossover like other community detection algorithms[5,32] to replace the recombination in NNIA. Select two arbitrary safe parent individuals, and then produce a random binary vector. If the vector is ***1*** then select the genes from the first parent, otherwise select the genes from the second parent and combine the genes to form the child. Because of the biased initialization, the child generated from the two safe parents is guaranteed to be safe. That is to say, if a gene i contains a value j, then the edge (i, j) exists.

In order to solve the problem of community detection using NNIA, the static hypermutation adopted in NNIA is also replaced by mutation operation suited to community detection. Thus we select the gene of the individual with a certain probability to mutate from the child population. However, the possible value of an allele must be one of the replaced gene's neighbors, which guarantees the mutated child is also safe as the crossover operation. The uniform crossover and mutation are shown in Fig.3.

Parent1	2	3	4	3	6	6	6
Parent2	4	3	2	1	6	5	5
Binary Vector	1	0	1	0	0	1	0
Offspring	2	3	4	1	6	7	5
Mutate Position			↑			↑	
New Offspring	2	3	2	1	6	5	5

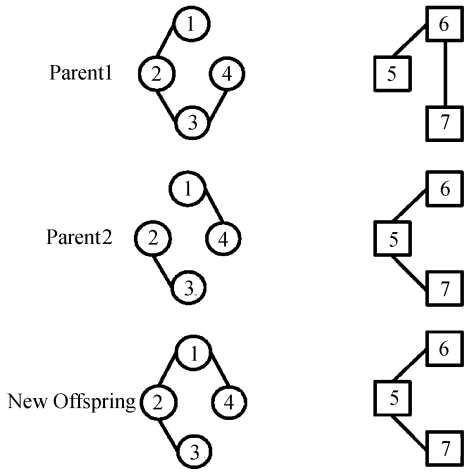

Fig.3. Illustration of the uniform crossover and mutation.

4.7 Local Search Strategy

In the opinion of Guimera and Amaral, when solving community detection problems, it is an effective method to generate a new candidate solution by continuously executing the following three types of operations on current candidate solution, which includes moving single nodes from one community to another, merging multi-communities and splitting single communities[34]. Crossover operator is regarded as a macroscopic operation on individuals, while the mutation operator is regarded as a microcosmic operation on individuals. Thus, if the crossover operator can achieve its global search function by merging and splitting communities, the mutation operator can achieve its local search function by moving single nodes between communities[35]. Inspired by this idea, our local search algorithm is based on the mutation operator. Because the local search strategy requires a single objective function, a weighted objective or a Tchebyscheff metric or any other metric which will convert multiple objectives into a single objective can be used. In our study, we use a weighted objective:

$$F(x) = \sum_{i=1}^{2} w_i f_i(x), \tag{9}$$

where w_1, w_2 are nonnegative weights for the two objectives, $f_i(x)$ are the objective functions which described in Subsection 4.1, the weights are calculated from the obtained set of solutions in a special way. First, the minimum $f_i^{\min}$ and maximum $f_i^{\max}$ value of each objective function f_i are noted. Thereafter, for any solution x in the obtained set, the weight for each objective function is calculated as follows[36]:

$$w_i = \frac{(f_i(x) - f_i^{\min})/(f_i^{\max} - f_i^{\min})}{\sum_{k=1}^{2}(f_k(x) - f_k^{\min})/(f_k^{\max} - f_k^{\min})}, \tag{10}$$

where the division of the numerator with the denominator ensures that the calculated weights are normalized or $\sum_{i=1}^{2} w_i = 1$.

However, in order to take advantage of the prior knowledge about relations between nodes, the mutation operator in the local search strategy is not randomly, but influenced by the neighbor nodes. There is an obvious intuition that the node will be in the same community with most of its neighbors. In other words,

if most of a node's neighbors are in the i-th community, the node will be in the i-th community with a larger probability. Therefore, in the mutation operation, we should find the labels of all the neighbors of the mutated node, and record the node label which the most neighbors owned. Then we randomly select one from these neighbor nodes to replace the original node which needs to mutate. It will not result in merging or splitting communities when move this node from one community to another. We will give the detailed procedure of the local search strategy in the following.

Algorithm 2. Local Search Procedure

Input: D_t (population before local search at the t-th generation),
S (size of dominant population),
K (number of neighbors).

Output: D'_t (improved population in the t-th generation).

Step 1: Set $i = 1$, $D'_t = \varnothing$.

Step 2: If $i > S$, the algorithm terminates. Export D'_t as the new population. Otherwise, select the i-th individual in D_t, set $k = 1$.

Step 3: If $k > K$, the search procedure stops for the i-th individual, adds the current individual to D'_t. Otherwise, go to Step 4;

Step 4: Assume the j-th gene need to do local search, attain all the neighbors of node j, find the label of community which most neighborhood nodes belonging to. And then select one from the nodes to replace node j by the corresponding value.

Step 5: Calculate the value of objective function of the new individual according to (10). If its value is greater than that before local search, add the new individual to D'_t, go to Step 7. Otherwise, go to Step 6.

Step 6: $k = k + 1$, go to Step 3.

Step 7: $i = i + 1$, go to Step 2.

4.8 Solution Selection

Actually, the algorithm DYN-LSNNIA returns a set of solutions at the end of each time step, which are all contained on the Pareto front. Each of these solutions corresponds to a different trade-off between the two objectives and thus to diverse partitioning of the network consisting of various number of clusters. The problem is how to select one best solution which denotes the optimal partitioning of the current network at each time step. A criterion should be established to automatically select one solution with respect to another. Unfortunately, there is still no effective selection method in current literature so far.

In this study, we employ the community score introduced in [32] that is proved very effective in detecting communities as the selection rule. The community score takes into account both the fraction of interconnections among the nodes and the number of interconnections contained in the module. It is defined as $CS = \sum_{i=1}^{k} score(\boldsymbol{C}_i)$, where

$$score(\boldsymbol{C}_i) = \frac{\sum_{i \in \boldsymbol{C}} \mu_i}{|\boldsymbol{C}|} \times \sum_{i,j \in \boldsymbol{C}} A_{ij}, \tag{11}$$

here $\mu_i = \frac{1}{|\boldsymbol{C}|} \sum_{j \in \boldsymbol{C}} A_{ij}$ denotes the fraction of edges connecting each node i of $\boldsymbol{C}$ to the nodes in the same community $\boldsymbol{C}$. The Community Score gives a global measure of the network division in communities by summing up the local score of each module found. The larger Community Score indicates the community structure is stronger. Thus the best solution we selected has the maximum Community Score in the set of solutions.

5 Experimental Study

In this section, we evaluate the effectiveness and efficiency of our algorithm on four synthetic datasets and two real-world networks. The compared algorithms include DYN-MOGA which is the only dynamic multiobjective community detection algorithm proposed by Folino *et al.*[5] and the DYN-LSNNIA without the local search strategy (termed as DYN-NNIA).

The parameter settings are as follows. In DYN-MOGA, crossover rate $p_C = 0.8$, mutation rate $p_M = 0.2$, elite reproduction equals 10% of the population size, and the selection strategy is roulette selection function. And the population size is 100, the number of generation is 300. For DYN-NNIA and DYN-LSNNIA, the maximum size of dominant population $n_D = 100$, the maximum size of active population $n_A = 20$, and the size of clone population $n_C = 100$, the crossover rate, mutation rate and the number of generation keep the same as DYN-MOGA. We conduct all the experiments on an Intel Core2 Duo 2.0 GHz PC with 1 GB of main memory, running on Windows XP. In the following experiments, the reported data are the statistical results based on 30 independent runs on each dataset.

5.1 Experiments on Synthetic Datasets

We generate synthetic datasets by following the procedure suggested by Newman and Girvan[7]. The data consists of 128 nodes that belong to 4 communities with 32 nodes in each community. Edges are placed independently and randomly between a pair of nodes, the probability that a link exists between a pair of nodes belonging to the same community is p_{in}; the probability that a link exists between a pair of nodes belonging to different communities is p_{out}. The value of p_{in} and p_{out} are chosen such that the average degree of each

node is set to 16. In order to control the noise level in the dynamic networks, a parameter z_{out}, which represents the mean number of edges from a node to nodes in other communities, is introduced to describe the synthetic datasets. If we increase the value of z_{out}, then p_{in} becomes smaller, p_{out} becomes larger, the network will become more noisy in the sense that the community structure becomes less obvious and hard to detect. In this study, we generate the datasets under four different noise levels by setting $z_{out} = 3, 4, 5, 6$. In order to introduce dynamics into the network, we let the community structure of the network evolve in the following way. At each time step after time step 1, we randomly choose 10% of the nodes to leave their original community and join the other three communities at random. After the community memberships are decided, links are generated by following the probabilities p_{in} and p_{out} as before. We generate the network with community evolution in this way for 10 time steps.

Since we have the ground truth answer for communities and their memberships at each time step, we can directly measure the accuracy of the clustering results. We adopt NMI described in Subsection 4.1 to measure the similarity between the true partitions and the detected ones. In order to evaluate the results dependably, we use the standard error of NMI at each time step to describe the stability of the algorithms. The standard error of a statistic is the standard deviation of the sampling distribution of that statistic. Standard errors are important because they reflect how much sampling fluctuation statistics will show. The inferential statistics involved in the construction of confidence intervals and significance testing are based on standard errors. The standard error of a statistic depends on the sample size. In general, the larger the sample size, the smaller the standard error. In our experiments, the sample size is 30.

Fig.4 shows the statistical average value of NMI with respect to the ground truth over the 10 time steps when $z_{out} = 3$. It can be seen that the average values of NMI at each time step obtained by both DYN-LSNNIA and DYN-NNIA equal 1, which illustrates these two algorithms can detect the true community structure at each time step. However, DYN-MOGA cannot always get the value 1 at each time step. In addition, the standard error obtained by DYN-MOGA at each time step is larger than those of both DYN-NNIA and DYN-LSNNIA. It can be known that the results got by DYN-MOGA are not steady enough compared to the other two algorithms.

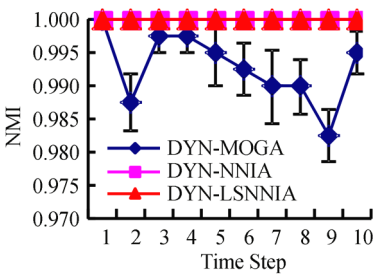

Fig.4. NMI when $z_{out} = 3$.

Fig.5 presents the community score obtained by three algorithms at each time step. The larger community score is obtained, which indicates the corresponding network is densely connected within each subnetwork. It can be seen that the community scores got by these three algorithms at each time step are almost the same, because the generated network can be detected effortlessly when $z_{out} = 3$.

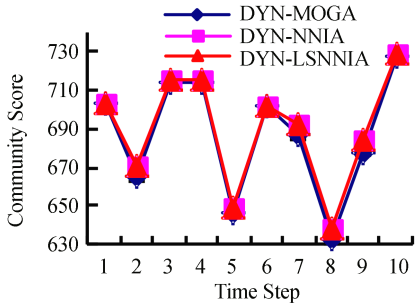

Fig.5. Community score when $z_{out} = 3$.

Figs. 6, 7, 8 illustrate the statistical average values of NMI over the 10 networks for the 10 time steps, when $z_{out} = 4, 5, 6$. It can be seen that the algorithm DYN-LSNNIA can still achieve very high accuracy compared

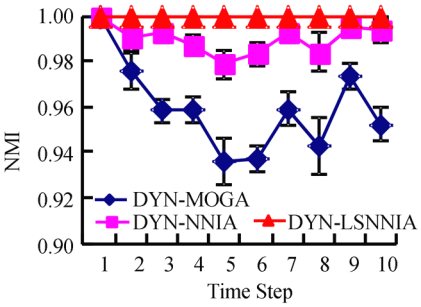

Fig.6. NMI when $z_{out} = 4$.

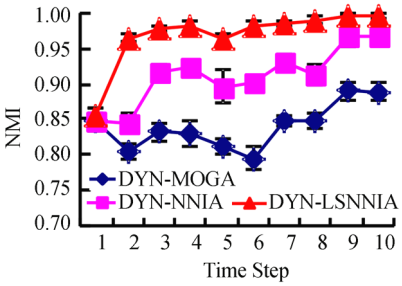

Fig.7. NMI when $z_{out} = 5$.

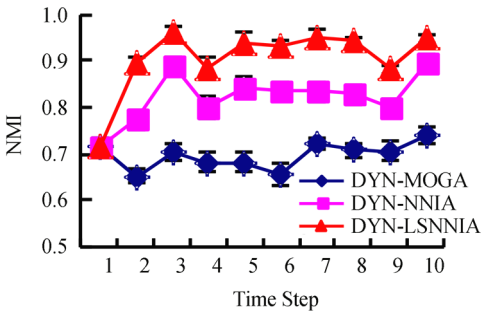

Fig.8. NMI when $z_{out} = 6$.

to the other two algorithms when the noise level becomes high. However, it is worth to notice that the value got by the three algorithms at time step 1 is basically equal. This is because that we use GA-Net to handle with the initial network in these three algorithms. Thus the results at time step 1 almost are the same. The same phenomena can be seen in all these experiments.

From Fig.6, we can see that only the DYN-LSNNIA can find the true community structure, while the other two algorithms fail. As can be seen from Figs. 6, 7 and 8, with the variation of noise level, the average value of NMI becomes smaller, which demonstrates the network becomes too complex to detect. Even so, our algorithm DYN-LSNNIA can still get the better results than the other algorithms. Moreover, the algorithm DYN-LSNNIA is the most steady of all the three algorithms, which can be seen from the standard error of the NMI at each time step.

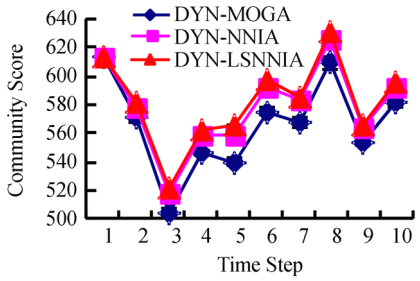

Fig.9. Community score when $z_{out} = 4$.

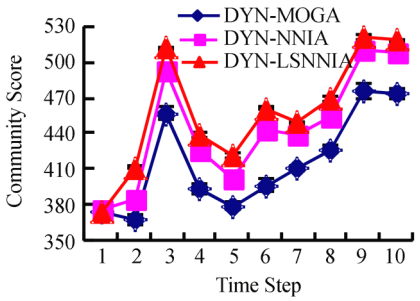

Fig.10. Community score when $z_{out} = 5$.

The community score obtained by the three algorithms when $z_{out} = 4$, 5, 6 are shown in Figs. 9, 10 and 11 respectively. We can find that the algorithm DYN-LSNNIA still outperforms the other two algorithms. That is to say, the solutions selected by our algorithm denote the results approaching to the true community structure.

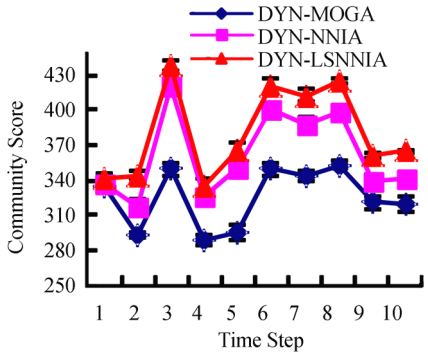

Fig.11. Community score when $z_{out} = 6$.

5.2 Experiments on Real-World Datasets

In this subsection, we present experimental studies on two real-world datasets: the football dataset (http://www.jhowell.net/cf/scores/scoresinde-x.htm) and the VAST dataset (http://www.cs.umd.edu/hcil/VAST challenge08/).

5.2.1 Football Dataset

The football dataset is the National Collegiate Athletic Association (NCAA) Football Division 1-A schedule data, which has been used by Newman[8]. The NCAA divides 116 schools into 11 conferences. In addition, there are 4 independent schools: Army, Brigham Young, Navy, and Notre Dame, where nodes represent teams and edges represent the regular season games between the two teams they connect. Each conference contains around 8∼12 teams. Games are more frequent between members of the same conference than between members of different conferences, with teams playing an average of about 7 intraconference games and 4 interconference games in each year[8]. In our study, we select the years 2005∼2009 to evaluate our algorithm, each year as one time step, where the number of conferences is 12 and the number of teams is 120. Because the community structure of the football data has been known, we still adopt NMI to evaluate our algorithm as before.

Fig.12 shows the statistical average value of NMI with respect to the ground truth over the 5 time steps. It also presents the better performance of our algorithm compared to the other two algorithms. The average value of NMI obtained by our algorithm is over 0.9

at each time step except time step 1, which illustrates our algorithm can discover the nearly true community structure at each time step. The community scores obtained by the three algorithms are shown in Fig.13.

From Figs. 12 and 13, we can see that the results got by the three algorithms are becoming better gradually over time except time step 3. Through our analysis, this is because the regular season games between these teams in 2007 are more frequent and irregular. However, this situation is improved from year 2008 to 2009, the regular season games between members of the same conference were arranged more. Thus, the community structure found in these two time steps can be more clear and accurate. In addition, we can also reach the similar conclusions as the synthetic datasets, the results obtained by our algorithm are the most steady of all three algorithms from the error bars shown in Figs. 12 and 13.

In order to analyze visually, we use Pajek software[37] to display the communities recognized by our algorithm DYN-LSNNIA on the football data for the year 2009 shown in Fig.14. The small circles with the same color denote the nodes in one community. We associated 12 distinct RGB colors with the 12 true communities which the teams really belong to.

From Fig.14, we can see that our algorithm can be able to recognize 11 different communities. Almost all

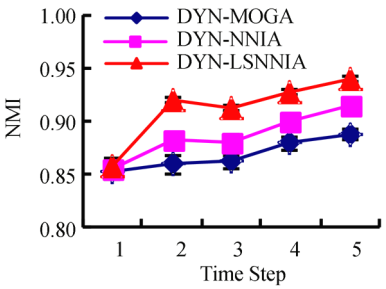

Fig.12. NMI of the football dataset.

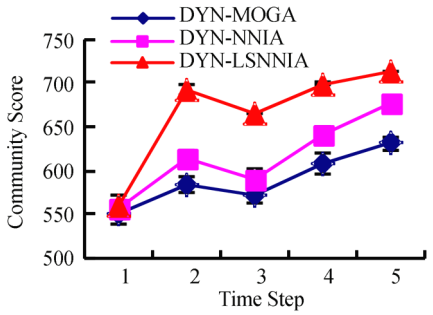

Fig.13. Community score of the football dataset.

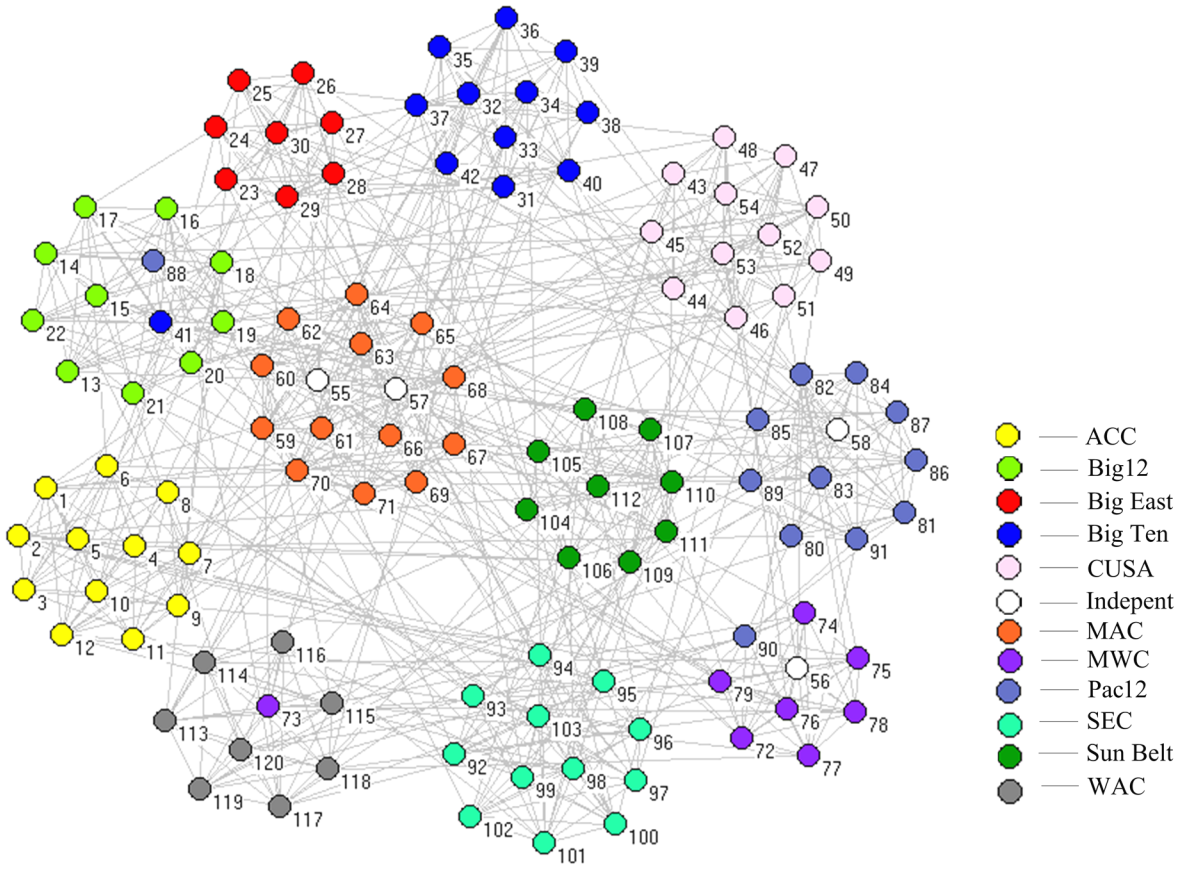

Fig.14. Communities found by DYN-LSNNIA on the football data for the year 2009.

teams are correctly grouped with the other teams in their conference, which is an impossible mission for the other two algorithms. Only several teams are mistakenly divided, which is shown in different colors in the 11 partitions. That is to say, only the conferences Big 12, MAC, MWC, Pac 12, and WAC have the incorrect teams, which are not belonging to these conferences originally. However, there are 4 independent teams that do not belong to any conference. They tend to be grouped with the conference with which they are most closely associated. In short, our algorithm achieves the best performance of all three algorithms.

5.2.2 *VAST Dataset*

The VAST dataset is a challenge task from IEEE VAST 2008. However, our experiment is only based on the VAST contest 2008 mini challenge 3, whose primary task is to characterize the Catalno/Vidro social network based on the cell phone call data provided and to characterize the temporal changes in the social structure over the 10-day period.

This dataset consists of information about 9 834 calls between 400 cellphones over a 10-day period in June 2006 in the Isla Del Sueno. It includes records with the following fields: identifier for caller, identifier for receiver, time, duration and call origination cell tower. In order to detect the communication patterns, we construct call graphs based on the call records. A call graph G is a pair (V, E), where V is a finite set of vertices (mobile users), and E is a finite set of vertex-pairs from V (mobile calls). So if user u calls user v, then an edge (u, v) is said to exist in E. We convert the input social network and the corresponding dynamic graph G into 10 different snapshot graphs.

Because we have no idea about the ground truth of the cellphone network, we use the modularity to evaluate the network. If the modularity of the network is larger than the other one, it indicates the network connected strongly. We only discover the hidden community structure in the network with our algorithm. Fig.15 shows the statistical average value of modularity of Catalno/Vidro social network over 10 time steps. It can be seen that our algorithm outperforms the other two algorithms at each time step except time step 1. Similar to the above results, the community structure found by our algorithm are not only densely connected, but also more steady at each time step. The community score obtained by the three algorithms are shown in Fig.16.

It is known that this is a challenge task from IEEE VAST 2008. Thus this dataset has been analyzed by many researchers. We have known that the structure of the cellphone network changed drastically on the 8th

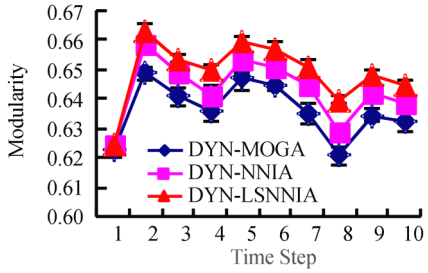

Fig.15. Modularity of VAST dataset.

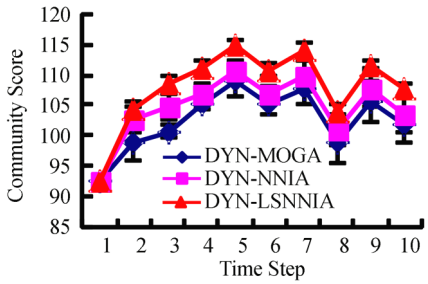

Fig.16. Community score of VAST dataset.

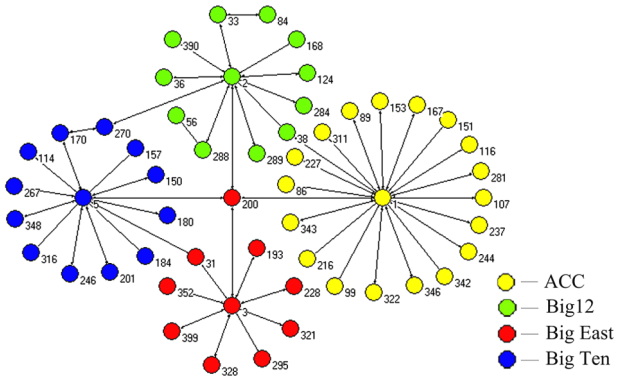

Fig.17. Main community structure of VAST found by DYN-LSNNIA at time step 7.

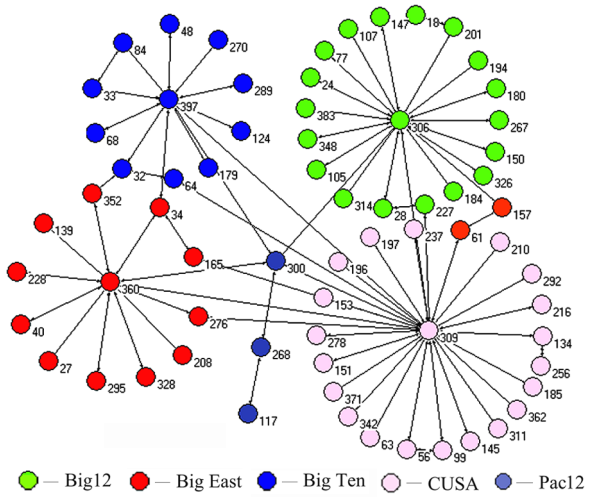

Fig.18. Main community structure of VAST found by DYN-LSNNIA at time step 8.

day[38]. In other words, there is a significant variation happened at the high-level leaders during this period. We display the main structure of the cellphone network at time step 7 in Fig.17 and time step 8 in Fig.18. From these two figures, we find that node 200 is the main boss while nodes 1, 2, 3, 5 are important nodes in the Catalano hierarchy at time step 7. While at time step 8 the nodes 300, 306, 309, 360, 397 emerge into prominence. The community structure discovered by our algorithm is consistent with the above analysis.

6 Concluding Remarks

In this paper, a novel multiobjective community detection algorithm is proposed to discover communities and capture community evolutions in dynamic social networks. Experimental results on synthetic datasets and real-world networks demonstrate that our algorithm can obtain the better performance than the two compared methods. It can achieve better accuracy in community extraction and capture community evolution more faithfully. The results obtained by the algorithm DYN-LSNNIA are not only more accurate, but also more steady than the other two algorithms. However, the time-consuming problem should be dedicated to in our future work. We will expand our algorithm for processing the large-scale networks in real life.

References

[1] Greene D, Doyle D, Cunningham P. Tracking the evolution of communities in dynamic social networks. In *Proc. Int. Conf. Advances in Social Networks Analysis and Mining*, August 2010, pp.176-183.

[2] Yang T B, Chi Y, Zhu S H, Gong Y H, Jin R. Detecting communities and their evolutions in dynamic social networks — a Bayesian approach. *Machine Learning*, 2011, 82(2): 157-189.

[3] Lin Y R, Chi Y, Zhu S H, Sundaram H, Tseng B L. Facetnet: A framework for analyzing communities and their evolutions in dynamic networks. In *Proc. the 17th Int. Conf. World Wide Web*, April 2008, pp.685-694.

[4] Chakrabarti D, Kumar R, Tomkins A. Evolutionary clustering. In *Proc. the 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, August 2006, pp.554-560.

[5] Folino F, Pizzuti C. A multiobjective and evolutionary clustering method for dynamic networks. In *Proc. Int. Conf. Advances in Social Networks Analysis and Mining*, August 2010, pp.256-263.

[6] Newman M E J, Girvan M. Finding and evaluating community structure in networks. *Physical Review E*, 2004, 69(2): 026113.

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

[8] Newman M E J. Fast algorithm for detecting community structure in networks. *Physical Review E*, 2004, 69(6): 066133.

[9] Duch J, Arenas A. Community detection in complex networks using extremal optimization. *Physical Review E*, 2005, 72(2): 027104.

[10] Fortunato S. Community detection in graphs. *Physics Reports*, 2010, 486(3-5): 75-174.

[11] Lancichinetti A, Fortunato S, Kertesz J. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Physics*, 2009, 11(3): 033015.

[12] Du N, Wang B, Wu B. Community detection in complex networks. *Journal of Computer Science and Technology*, 2008, 23(4): 672-683,

[13] Kumar R, Novak J, Raghavan P, Tomkins A. On the bursty evolution of blogspace. In *Proc. the 12th Int. Conf. World Wide Web*, May 2005, pp.568-576.

[14] Leskovec J, Kleinberg J, Faloutsos C. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proc. the 11th Int. Conf. Knowledge Discovery and Data Mining*, August 2005, pp.177-187.

[15] Palla G, Barabasi A L, Vicsek T. Quantifying social group evolution. *Nature*, 2007, 446(7136): 664-667.

[16] Asur S, Parthasarathy S, Ucar D. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data*, 2009, 3(4): Article No. 16.

[17] Sarkar P, Moore A W. Dynamic social network analysis using latent space models. *ACM SIGKDD Exploration Newsletter*, 2005, 7(2): 31-40.

[18] Chi Y, Song X D, Zhou D, Hino K, Tseng B L. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proc. the 13th Int. Conf. Knowledge Discovery and Data Mining*, August 2007, pp.153-162.

[19] Ahmed A, Xing E. Dynamic non-parametric mixture models and the recurrent Chinese restaurant process: With applications to evolutionary clustering. In *Proc. the 8th SIAM Int. Conf. Data Mining*, April 2008, pp.219-230.

[20] Tang L, Liu H, Zhang J, Nazeri Z. Community evolution in dynamic multi-mode networks. In *Proc. the 14th Int. Conf. Knowledge Discovery and Data Mining*, August 2008, pp.677-685.

[21] Kim M S, Han J W. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc. Very Large Data Base Endowment*, 2009, 2(1): 622-633.

[22] Kim K, McKay R, Moon B R. Multiobjective evolutionary algorithms for dynamic social network clustering. In *Proc. the 12th Conf. Genetic and Evolutionary Computation*, July 2010, pp.1179-1186.

[23] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evolutionary Computation*, 1999, 3(4): 257-271.

[24] Knowles J, Corne D. Approximating the non-dominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 2000, 8(2): 149-172.

[25] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 2002, 6(2): 182-197.

[26] Coello C C A, Pulido G T, Lechuga M S. Handing multiple objectives with particle swarm optimization. *IEEE Trans. Evolutionary Computation*, 2004, 8(3): 256-279.

[27] Zhang Q F, Zhou A M, Jin Y. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans. Evolutionary Computation*, 2008, 12(1): 41-63.

[28] Zhang Q F, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evolutionary Computation*, 2007, 11(6): 712-731.

[29] Gong M G, Jiao L C, Du H F, Bo L F. Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary Computation*, 2008, 16(2): 225-255.

[30] Danon L, Daz-Guilera A, Duch J, Arenas A. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005, 2005(9): P09008.
[31] Park Y J, Song M S. A genetic algorithm for clustering problems. In *Proc. the 3rd Conf. Genetic Programming*, July 1998, pp.568-575.
[32] Pizzuti C. GA-Net: A genetic algorithm for community detection in social networks. In *Proc. the 10th Int. Conf. Parallel Problem Solving from Nature*, September 2008, pp.1081-1090.
[33] Cormen T H, Leiserson C E, Rivest R L, Stein C. Introduction to Algorithms (2nd edition). Cambridge: MIT Press, 2001.
[34] Guimera R, Amaral L A N. Functional cartography of complex metabolic networks. *Nature*, 2005, 433(7028): 895-900.
[35] Jin D, He D X, Liu D Y, Baquero C. Genetic algorithm with local search for community mining in complex networks. In *Proc. the 22nd Int. Conf. Tools with Artificial Intelligence*, October 2010, pp.105-112.
[36] Deb K, Goel T. A hybrid multi-objective evolutionary approach to engineering shape design. In *Proc. the 1st International Conference on Evolutionary Multi-Criterion Optimization*, March 2001, pp.385-399.
[37] de Nooy W, Mrvar A, Batagelj V. Exploratory Social Network Analysis with Pajek. New York: Cambridge University Press, 2005.
[38] Ye Q, Zhu T, Hu D Y, Wu B, Du N, Wang B. Cell phone mini challenge award: Social network accuracy — Exploring temporal communication in mobile call graphs. In *Proc. IEEE Symp. Visual Analytics Science and Technology*, October 2008, pp.207-208.

Mao-Guo Gong received the B.Eng. degree in electronic engineering and Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively. Since 2006, he has been a teacher with Xidian University. In 2008 and 2010, he was promoted as an associate professor and a full professor, respectively, both with exceptive admission. He is currently a full professor with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University. His research interests include computational intelligence with applications. Dr. Gong is a senior member of the China Computer Federation, a member of IEEE and ACM, and an executive committee member of the Natural Computation Society of Chinese Association for Artificial Intelligence. He was the recipient of the New Century Excellent Talent in University of the Ministry of Education of China, the 8th Young Scientist Award of Shaanxi, the New Scientific and Technological Star of Shaanxi Province.

Ling-Jung Zhang received the Bachelor's degree in electronic information engineering from Henan Normal University in 2009. He is currently pursuing the Master degree at Xidian University. His research is dedicated to community detection in dynamic networks.

Jing-Jing Ma received the Bachelor's degree in electronic engineering from Xidian University in 2004. She is currently pursuing the Ph.D. degree at Xidian University. Her research interests are mainly in the area of computational intelligence and image understanding.

Li-Cheng Jiao received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. Since 1992, Dr. Jiao has been a professor in the School of Electronic Engineering at Xidian University. Currently, he is the dean of the School of Electronic Engineering and the director of the Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China at Xidian University. His research interests include image processing, natural computation, machine learning, and intelligent information processing.