Selecting Multiple Network Spreaders based on Community Structure using Two-Phase Evolutionary Framework

Yu-Hsiang Fu
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, ROC
yuhsiangfu.cs98g@nctu.edu.tw

Chung-Yuan Huang
Department of Computer Science
and Information Engineering
Chang Gung University
Taoyuan, Taiwan, ROC
gscott@mail.cgu.edu.tw

Chuen-Tsai Sun
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, ROC
ctsun@nctu.edu.tw

***Abstract*—The identification of multiple network spreaders is an appropriate solution to spread information, ideas or diseases in many practical applications. For instance, in target marketing, the spreaders are selected from customer groups classified by similar purchase behaviors to advertise the products, and to optimize the allocation of limited resources. The community detection approaches intuitively are used to identify the community structures or social groups in a social/complex network. However, how to determine the number of community *K* is a difficult issue. Hence, two-phase evolutionary framework (TPEF) is proposed for automatically determining the number of community *K* and maximizing the modularity of communities. In the preliminary experiment, the LFR benchmark networks are used to test the proposed method, and to analyze the execution time, the community quality and the network spreading effect. The experiment results show that TPEF can perform well and produce the satisfied quality of community structures. The community detection approaches can be used to assist selecting the multiple network spreaders, and to gain the benefit in network spreading when the community structure is obvious. Furthermore, our results suggest that developing an index, a mechanism or a sampling technic is necessary to decide whether the community detection approaches are applied for selecting multiple network spreaders.**

Keywords—genetic algorithm; community detection; network spreading; social network analysis; multiple network spreaders.

I. INTRODUCTION

Identifying the most influence network spreaders can accelerate or hinder spreading information, ideas and diseases in a social/complex network [1]-[5]. The correspondence strategy also can be built by the identified spreaders to increase exposure range of product in marketing, to detect the contagious outbreak and to execute the early intervention in disease [6], and to speed up the spreading in information dissemination or diffusion on Internet [2], [4], [5]. Hence, how to identify the influence network spreaders becomes an important issue for many research domains. In the social network analysis, the centrality measurements are used to evaluate the importance of nodes by analyzing the network structure. These centrality measurements can be classified into local centrality and global centrality categories [7], [8].

The degree centrality is a simple and effective local centrality to measure the importance of nodes [7], [8]. A node with high degree has more connections and higher influence in the network, e.g. a hub node with a lot of connections. On the other hand, the betweenness, closeness and *k*-shell decomposition are means to measure global centralities [1], [3], [7], [8]. High betweenness means a node is located on many shortest paths in communication, and high closeness means a node is a network center with shorter average length of shortest paths from the node to the other nodes in the network, and high *k*-shell values means a node is located within the core layer of network analyzing by *k*-shell decomposition method. In addition, the diffusion based approach, the PageRank algorithm is also applied to measure the importance of nodes in the network [9]. High PR value means a node is connected with many important neighbors. Also, the method of considering global diversity and local features is robust and less-sensitive according to the simulation results of network spreading [7].

However, the centrality methods are not sufficient for identifying the multiple network spreaders as an appropriate solution in many practical applications. For instance, in target marketing, the spreaders are selected from customer groups classified by similar purchase behaviors to advertise the products, and to optimize the allocation of limited resources; in disease prevention, the spreaders are chosen from social groups organized by different social behavior to execute the respective interventions in the contact network. Hence, the community detection approach intuitively is used to identify the community structures or social groups in the network for selecting multiple network spreaders [8][10]-[15]. Furthermore, how to select multiple network spreaders is discussed in the recent research works. Borgatti defined the key-players problem (KPP) as well as selecting multiple network spreaders by two aspects [16]: (1) KPP-Pos is a set of key-players which are maximally connecting to all others nodes; (2) KPP-Neg is a set of key-players which are removed causing the residual network with the least possible cohesion. Kitsak argued that the distance between the spreaders should be considered to determine the extent of overlap in network spreading [3].

According to Borgtti's KPP-Pos definition, the problem of finding a set of key-players (or limited number of key-players) can be reduced to the set (or vertex) cover problem or 0/1

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

Knapsack problem. In addition, the community detection problem is to identify the community structure in the network; then, the network spreaders can be found from the communities as represented centers. Therefore, the network community detection problem can be reduced to the graph partition problem. However, the above reduced problems are NP-Complete (NPC) problems [17]. The evolutionary computing approaches, e.g. genetic algorithm (GA) [17]-[19], are suitable used to find the approximate solution of a NPC problem. The advantages of applying genetic algorithm are (1) the approximate solution can be found efficiently under the specific limitations; (2) the content of approximate solution can be analyzed, understood and preserved, e.g. the information of community structure can be extracted from the chromosomes, then stored into files.

In this study, the community structures of a network are used as a mean of selecting *K* network spreaders as well as *K* key-players based on the concepts of KPP-Pos definition, resources limitation (i.e. limited number of network spreaders) and community detection. The criteria of identifying community structures is maximizing the modularity [8], [10]-[12]. Also, the condition of selecting network spreaders is that the nodes have the most intra-community connections and the least inter-community connections [16]. The selected network spreaders are expected to spread information, ideas and virus from the inside to the outside of communities in the network. However, how to determine the number of community *K* of the given network is a difficult issue, because the number of community *K* usually is unknown. Hence, for the purpose of automatically determining the number of community *K*, the two-phase evolutionary framework (TPEF) is proposed as a trade-off between the performance efficiency and the compactness of identified communities. In the phase one, automatically determine an appropriate number of community *K* based on sampling of network topology. In the phase two, optimize the network partition produced by the phase one. Finally, the multiple network spreaders are chosen from the identified communities.

In the preliminary experiment, the LFR benchmark model [20], [21] is used to test the partition-based GA (i.e. standard GA, SGA) [22], the topology-based GA (i.e. locus-based GA, LGA) [23] and the TPEF by analyzing the execution time and the quality of community structures. The experiment results show that TPEF can determine an appropriate number of community *K* and produce the satisfied community structures. In the simulation results of network spreading, for the case of distinct community structure (i.e. $u < 0.2$), we found that using community detection approach to select network spreaders can reach much wider spreading range than centrality methods. However, in the case of indistinct community structure (i.e. $u \geq 0.2$), using centrality based methods to select *K* network spreaders can performs well, and network spreading results are similar to the community detection approach. In other words, there is a benefit of applying community detection methods to assist the selecting *K* network spreaders in the network with the distinct community structures. Otherwise, the centrality methods could be used to select the network spreaders, e.g. degree centrality. The experiment results also indicate that an index, a mechanism or a sampling technic is needed to decide whether the community detection approach should be applied to gain the benefit of community structures in network spreading.

II. Background

To present a social/complex network, let an undirected graph $G = (V, E)$, where V is the node set and E is the edge set of network. Let $|V|$ indicates the number of nodes and $|E|$ indicates the number of edges. The network structure is represented as an adjacency matrix $A = \{a_{ij}\}$ and $a_{ij} \in R^n$, where $a_{ij} = 1$ if a link exists between node *i* and *j*, otherwise $a_{ij} = 0$.

A. Local and Global Centralities

Degree centrality is a simple yet effective method for measuring node influence in a network [7], [8]. Let $C_d(i)$ denote the degree centrality of node *i*. A high degree centrality indicates a large number of connections between a node and its neighbors. $NB_h(i)$ denotes the set of neighbors of node *i* at a *h*-hop distance. The degree centrality of node *i* is therefore defined as

$$C_d(i) = |NB_h(i)| = \sum_{j=1}^{n} a_{ij} \tag{1}$$

where $|NB_h(i)|$ is the number of neighbors of node *i* at the *h*-hop distance; in most cases, $h = 1$.

Betweenness centrality measures the proportion of the shortest paths going through a node in a network [7], [8]. Let $C_b(i)$ denote the betweenness centrality of node *i*. A high betweenness value indicates that a node is located along an important communication path. Accordingly, the betweenness centrality of node i is defined as

$$C_b(i) = \sum_{s \neq t \neq v \in V} \frac{|Q_{st}(i)|}{|Q_{st}|} \tag{2}$$

where $|Q_{st}(i)|$ is the number of shortest paths from node *s* to node *t* through node *i*, and $|Q_{st}|$ the total number of shortest paths from node *s* to node *t*.

Closeness centrality measures the average length of the shortest paths from one node to other nodes [7], [8]. Let $C_l(i)$ denote the closeness centrality of node *i*. A high closeness centrality value indicates that a node is located in the center of a network, and that the average distance from that node to other nodes is shorter compared to nodes with low closeness centrality. The closeness centrality of node *i* is defined as

$$C_l(i) = \frac{1}{\overline{l_i}}, \overline{l_i} = \frac{1}{n} \cdot \sum_{j=1}^{n} l_{ij} \tag{3}$$

where $\overline{l_i}$ is the average length of the shortest paths from node *i* to the other nodes, and l_{ij} is the distance from node *i* to node *j*.

B. k-shell Decomposition and PageRank

The *k*-shell decomposition [1], [3] iteratively assigns a *k*-shell index value to every node in a network. During the first step let $k = 1$, and remove all nodes where $C_d(i) = k = 1$. After removal, the degrees of some remaining nodes may be $k = 1$. Nodes are continuously pruned from the network until

there are no $k = 1$ nodes. All removed nodes are assigned a *k*-shell value of $ks = 1$. The similar procedure is continued until all network nodes are removed and assigned a *k*-shell index value.

PageRank (PR) is a diffusion-based measurement to evaluate the importance of nodes, and to describe a random walk process in a network structure; PageRank is also an instance in the class of eigenvector centrality method [4], [5], [9]. PageRank is initially proposed to rank web pages in the World Wide Web (WWW). High PR value means that a web page is linked by other pages of high PR value. The PR value of node *i* is defined as

$$PR(i) = \frac{1-d}{n} + d \cdot \sum_{j \in NB_{h=1}(i)} \frac{PR(j)}{C_d^{out}(j)} \qquad (4)$$

where $PR(i)$ is the PR value of node *i*, $C_d^{out}(j)$ is the out-degree of neighbor *j*. d is a damping factor which means the probability d of a random walker would follow the link structure; otherwise, the probability $1 - d$ of a random walker would randomly jump to other web pages; d is usually assigned 0.85 as well as used in this work.

C. Community Detection and Modularity

In the community detection, the goal is to identify the best network partition which the edge connectivity is compact inside the community, and is loose between communities [10]-[13]. The variation of network type makes different meaning of the identified community structure corresponding to a set of web pages of the same topic in WWW, to a social group of high school, college, institute or company in an acquaintance network, to a circuit, pathway or motif of a certain synthesizing or regulating function in a metabolic network, and to a target group of customer who has the same conventional purchase behavior in a social network.

However, there are many possible network partitions for a given network; how to measure the quality of the network partition becomes an important issue. Girvan and Newman propose an edge betweenness based hierarchical clustering method to identify the community structures [10], and a modularity measure to evaluate the quality of the identified community structures [11], [12]. In the modularity, a meaningful network partition is that many edges inside communities and only few edges between communities. For a network with *K* communities, the modularity is defined as

$$Q = \sum_{i=1}^{K} (e_{ii} - a_i^2) = \sum_{i=1}^{K} \left(\frac{p_i}{|E|} - \left(\frac{q_i}{2|E|} \right)^2 \right) \qquad (5)$$

where e_{ii} is the fraction of edges that two endpoints are within the community m_i, a_i is the fraction of edges that at least one endpoint is within the community m_i. Moreover, p_i is the number of edges which two endpoints are within the community m_i, q_i is the sum of degree of community m_i as a node.

D. Community Detection using GA

According to the problems of selecting multiple network spreaders (or *K* key-players) and community detection can be reduced to the NPC problems [17]. In general, the evolutionary computing methods, e.g. genetic algorithm, are suitable used to find the approximate solution instead of optimal solution. Furthermore, single objective [24]-[26] or multiple objectives [27]-[31] GA have been applied to the community detection problem; the related gene coding representations (e.g. straight-forward representation [24] and locus-based representation [23]) and genetic operations have been proposed; the modularity and community score have been used as fitness function in GA.

The gene representation of GA consists of encode and decode steps (or called genotype and phenotype). For example, a network with *K* communities, each solution of network partition is represented as an individual (or a chromosome) $idv = \langle g_1, g_2, \dots, g_n \rangle$ in the population $pop = \{idv_1, idv_2, \dots, idv_n\}$. In the encode step of straight-forward representation (as Fig. 1), every gene is simply assigned a community id value $g_i \in [1, K]$ which means the node *i* belongs to community m_i. In the decode step, the community structure is reconstructed by decoding the information encoded in an individual, and every node *i* is simply classified into community m_i by the gene value g_i. On the other hand, in the encode step of locus-based representation, each gene is randomly assigned a neighbor id $g_i \in NB_{h=1}(i)$ which means an edge exists between node *i* and node *j*. In the decode step, an additional transformation approaches are used to reconstruct the community structure of a network, e.g. Union-Find algorithm and Disjoint-Set data structure [17].

In the related literatures of genetic operations, the selection operation can be roulette wheel selection, tournament selection, truncation selection, elitist selection [22], [27], etc. The operations of crossover and mutation should be depending on which the gene representation is used. In the straight-forward representation, the crossover operation could be one-way (or two-way) crossover [24]-[26]; because the traditional one-point (or multi-points) crossover could destroy the community structure encoded in an individual. In the one-way crossover, two individuals, i.e. idv_{src} and idv_{dest}, are randomly chosen from population pop. A gene g_i of idv_{src} is randomly selected and decide its community id value *i*, and all genes which belong to community m_i are duplicated to the same position (i.e. allele)

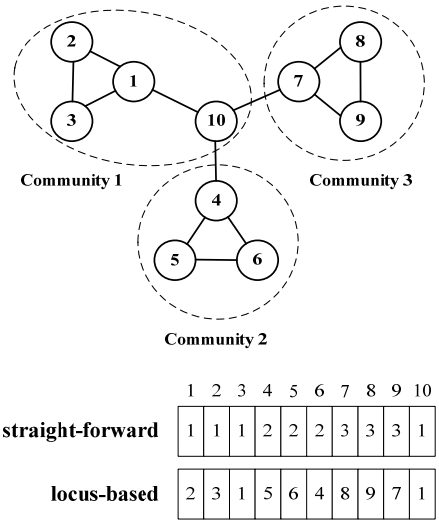

Fig. 1. Straight-forward and locus-based gene representations.

in idv_{dest}. The mutation operation could be bit-by-bit mutation or Self-Evolution (SE) method [24]. On the other hand, in the locus-based representation, the crossover operation could be uniform crossover; the mutation operation is similar to bit-by-bit mutation but with a constraint that every gene is randomly assigned a neighbor id based on the mutation rate [23], [28]-[31].

III. THE PROPOSED FRAMERWORK

The purpose of two-phase evolutionary framework (TPEF) is giving the considerations to the appropriate number of community *K* in a network, to the satisfactory performance efficiency, and to the quality of identified community structures. However, the performance efficiency and the quality of identified community structures might be a trade-off. For a given network with *K* communities, in the SGA, the simple gene representation makes good performance efficiency, but, large search space of solution (i.e. *K* options for each gene) might cause the poor quality of community structure. Another aspect, in the LGA, the topology-based gene presentation makes smaller search space of solution (i.e. few options in average are taken from neighbor) and better quality of community structure; and the number *K* can be determined automatic. Hence, TPEF takes the advantages from SGA and LGA.

For a given network, the number of community *K* is assumed unknown. The TPEF procedure is described as algorithm 1. First, initialize the variables of each phase, e.g. number of evolution num_{evo}, number of generation num_{gen}, number of population num_{pop}, selection rate $rate_{sel}$, crossover rate $rate_{cro}$ and mutation rate $rate_{mut}$. Second, in the phase one, LGA is used to determine the number of community *K* with appropriate quality of community structure; then the phase one returns the best individual idv_{best}^{p1}. Next, in the phase two, SGA is used to maximize the modularity of idv_{best}^{p1}; and the best individual $indv_{best}$ is kept by comparing the current best individual idv_{best}^{p2} and the previous $indv_{best}$. Final, TPEF returns the best individual $indv_{best}$ of a given network as the approximate solution of community detection.

The configuration of each phase in TPEF is described in Table 1, e.g. gene representation, genetic operation and fitness function. In phase one, the locus-based representation is used, each gene of an individual is randomly assigned by selecting neighbors' node id, and the Union-Find algorithm and Disjoint-Set data structure [17] are used as the additional transformation approach. In phase two, the straight-forward representation is used, each gene of an individual randomly assign a value which is picked from the distribution of neighbors' community id. In the truncation selection, the top-% chromosomes are preserved to create new individuals of population based on the selection rate $rate_{sel}$. In the uniform crossover, two parent individuals are randomly chosen from the population, and new children individuals are produced by exchanging each gene according to the crossover rate $rate_{cro}$. The bit-by-bit mutation is slightly different in each phase of TPEF; in the phase one, each gene is mutated based neighbors' node id; in the phase two, each gene is mutated based on the distribution of neighbors' community id. In the fitness function, the modularity, as in (5), is used and calculated by using algorithm 2.

TABLE I. THE CONFIGURATION OF TPEF.

Method	TPEF	
	Phase1	***Phase2***
Encoding	Locus-based	Straight-forward
Selection	Truncation selection	
Crossover	Uniform crossover	
Mutation	Bit-by-bit mutation	
Fitness	Modularity	

Algorithm 1: Two-Phase Evolutionary Framework

Input: network G, TPEF parameters num_{evo}, num_{gen}^{p1}, num_{pop}^{p1}, $rate_{sel}^{p1}$, $rate_{cro}^{p1}$, $rate_{mut}^{p1}$, num_{gen}^{p2}, num_{pop}^{p2}, $rate_{sel}^{p2}$, $rate_{cro}^{p2}$, $rate_{mut}^{p2}$.
Output: the best individual overall evolution idv_{best}.

1: $idv_{best} \leftarrow \{\}$
2: *InitialPhase1Variables*(num_{gen}^{p1}, num_{pop}^{p1}, $rate_{sel}^{p1}$, $rate_{cro}^{p1}$, $rate_{mut}^{p1}$)
3: *InitialPhase2Variables*(num_{gen}^{p2}, num_{pop}^{p2}, $rate_{sel}^{p2}$, $rate_{cro}^{p2}$, $rate_{mut}^{p2}$)
4: **For** *i*=1 to num_{evo}:
5: $idv_{best}^{p1} \leftarrow$ *Phase1DetermineClusterNumber*(G)
6: $idv_{best}^{p2} \leftarrow$ *Phase2MaximizeModularity*(G, idv_{best}^{p1})
7: **If** idv_{best} is empty:
8: $idv_{best} \leftarrow idv_{best}^{p2}$
9: **If** idv_{best}^{p2} is better than idv_{best}:
10: $idv_{best} \leftarrow idv_{best}^{p2}$
11: **Return** idv_{best}

The algorithm 2 used in each phase of TPEF is to calculate the modularity as the fitness value of every individual in the population. First, create a community set $M = \{m_1, m_2, \dots, m_K\}$ which is reconstructed by using the additional transformation approaches to decode the information of network structure of an individual, and $m_i = \{v_1, v_2, \dots, v_n\}$ is the set of nodes which belong to the community m_i where $v_j \in V$. Second, check the community set M whether contains the empty set after mutation operation. Next, create e_{intra} array of length *K* for counting and checking every edge $e_{ij} \in E$ whether the two-endpoints are inside the same community m_i. Final, calculate the modularity value as fitness of idv, as in (5).

The selection of multiple network spreaders is based on the identified community structure of TPEF. Each spreader is a represented center T_i and selected from nodes of the community m_i, and is defined as

$$T_i = \arg\max_{v_j \in m_i} C_{m_i}^r(j) \tag{6}$$

$$C_{m_i}^r(j) = \frac{|NB_{h=1}(j) \cap m_i|^2}{C_d(j) \times |m_i|} \tag{7}$$

where $C_{m_i}^r(j)$ is the center degree of node *j* in the community m_i. If $C_{m_i}^r(j)$ equals to 1 which means the node *j* are connecting

Algorithm 2: Fitness Function

Input: network G, individual idv.
Output: fitness value (modularity) of idv.

$Q \leftarrow 0$
$M \leftarrow CreateCommunitySet(idv)$
$K \leftarrow |M|$
For m_i in M:
  If m_i is empty:
    Return Q
$e_{ii} \leftarrow [0_1, 0_2, \ldots, 0_K]$
For $e = (v_s, v_t)$ in E:
  $c_s \leftarrow 0$
  $c_t \leftarrow 0$
  For i=1 to K:
    If v_s in m_i:
      $c_s \leftarrow i$
    If v_t in m_i:
      $c_t \leftarrow i$
  If $(c_s == c_t)$ and $\big((c_s > 0)$ and $(c_t > 0)\big)$:
    $e_{ii}[c_s] \leftarrow e_{ii}[c_s] + 1$
For i=1 to K:
  $p_i \leftarrow e_{ii}[i]/|E|$
  $q_i \leftarrow 0$
  For v_j in m_i:
    $q_i \leftarrow q_i + C_d(j)$
  $q_i \leftarrow (q_i/2|E|)^2$
  $Q \leftarrow Q + (p_i - q_i)$
Return Q

to all nodes of community m_i; otherwise, $C^r_{m_i}(j)$ is close to 0 which means the node j are connecting to the nodes of other communities in a network.

IV. Preliminary Experimental Results and Discussion

In the preliminary experiment, the LFR model [20], [21] is used to generate the synthesized networks with different property of community structure, and to test how well the quality of identified community structures is by the proposed algorithm. The degree distribution and community size distribution are assumed following the power-law in the LFR model. The parameters of LFR model are as follows: γ is the exponent of degree distribution; β is the exponent of distribution of community size; k_{max} and k_{min} are the upper-bond and lower-bond of node degree; z_{max} and z_{min} are the constraints of the community size; the mixing parameter u is a proportion of a node sharing links with the nodes of other communities, and $1 - u$ is a proportion of a node sharing the links with the nodes of its community. The parameter values of LFR model used in this study is shown in the Table 2. For each u, one synthesized network is generated to perform 20 rounds of the experiment.

A. Normalized Mutual Information

The normalized mutual information (NMI) is used to measure the quality of the identified community structures by the algorithms [32]. The similarity of true partition X and the identified partition Y is calculated in the NMI. The NMI is defined as

TABLE II. The Parameters used in the LFR model.

Parameters	Values
$\|V\|$	300
γ	2
β	1
u	0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6
$\langle k \rangle$	15
k_{max}	50
z_{max}	50
z_{min}	20

$$NMI(X,Y) = \frac{2 \cdot I(X,Y)}{H(X) + H(Y)} \tag{8}$$

$$I(X,Y) = \sum_x \sum_y P(x,y) log \frac{P(x,y)}{P(x)P(y)} \tag{9}$$

where $H(X) = -\sum_{x \in X} P(x) log P(x)$ is the Shannon entropy of true partition X, and $P(x)$ is the proportion of community x in the partition X. $H(Y)$ is also the Shannon entropy of identified partition Y, and $P(y)$ is the proportion of community y in the partition Y. $I(X,Y)$ is the mutual information of X and Y that means how well the Y can be found by the algorithm when X is given. $P(x,y)$ is the proportion of overlap of X and Y. If $NMI(X,Y)$ equals to 1 which means the partitions are identical, otherwise the partitions are independent.

B. Network-based Spreading Simulation

In this work, the SIR model and network-based spreading simulation (as algorithm 3) [3], [4], [7], [15], [33], [34] are used to evaluate the spreading ability of multiple network spreaders scenario based on the identified community structure of the algorithms. The SIR model consists of three states: susceptible (S), infective (I), and recovered (R). S set nodes are susceptible to information or diseases, I set nodes are capable of infecting neighbors, and R set nodes are immune and cannot be reinfected. Let $|S_t|$ denote the number of susceptible nodes at time t, $|I_t|$ the number of infected nodes at time t, $|R_t|$ the number of recovered nodes at time t, and $\rho[t]$ the proportion of immune nodes at time t. The total number of nodes in the SIR model is $|S_t| + |I_t| + |R_t| = |V|$. In the network-based spreading simulation, the step one, all nodes are in the S state except for the initial spreaders which are in the I state. The step two, each node in the I state randomly infects its neighbors according to an infection rate $rate_{inf}$ at each time step t, then enters the R state (i.e. recovery rate $rate_{rec} = 1$). The cumulative incidence of contagion $\rho[t] = |R_t|/|V|$ is calculated at the end of each time step. Repeat step 2 until the maximum time step requirement is satisfied—or, if necessary, when the I state set is empty.

C. Experiment Setting and Analysis Results

The hardware used for experiment environment is Intel Core 2 Quad Q8200S @ 2.33GHz CPU and 4 GB DDR4 RAM. The operating system is Microsoft Windows 7 Enterprise 64-bit SP1. The programs of SGA, LGA and TPEF are built by using Python 3.3, NetworkX 1.8.1 and Scipy-Stack-14.5.30. The analysis

Algorithm3: Network-based Spreading Simulation

Input: network G, time-step num_{step}, initial nodes I_{init}, infection rate $rate_{inf}$, recovery rate $rate_{rec}$.

Output: network spreading result ρ.

$S \leftarrow \{\} \cup V$
$I \leftarrow \{\}$
$R \leftarrow \{\}$
$\rho \leftarrow \left[0_0, 0_1, \ldots, 0_{num_{step}}\right]$
For t=0 to num_{step}:
  $I_t \leftarrow \{\}$
  $R_t \leftarrow \{\}$
  IF t is 0:
    $I \leftarrow I \cup I_{init}$
    $S \leftarrow S - I$
  Else:
    $I_t \leftarrow SusceptibleToInfected(G, S, I, rate_{inf})$
    $R_t \leftarrow InfectedToRecovered(I, R, rate_{rec})$
    $R \leftarrow R \cup R_t$
    $I \leftarrow I \cup I_t$
    $I \leftarrow I - R_t$
    $S \leftarrow S - I_t$
  $\rho[t] \leftarrow |R_t|/|V|$
Return ρ

results of experiments are execution time, community detection and network-based spreading simulation with multiple network spreaders scenario.

First, in the execution time analysis, the average running time of each method is calculated over 20 evolutions for each LFR network, and the setting of evolutionary algorithms is shown in table 3. In Fig. 2, the execution time of TPEF outperforms SGA and LGA, because the parameters num_{gen} and num_{pop} are set as a half of SGA and LGA; the performance efficiency is more important than the quality of community structure according to how much quick the appropriate number of community K can be known. The modularity of TPEF is the best when $u = 0.01$~0.1, is the second best when $u = 0.2$, and is very close to SGA when $u = 0.3$~0.6; all modularity results of TPEF are better than the results of LGA. Furthermore, the setting of each phase of TPEF can be adjusted according to the requirements, or depending on the number of community K is known (or unknown). For example, for the case of determining the number of community K, the parameters of phase one of TPEF can be set the same as LGA; for the case of optimizing the modularity, the parameters of phase two of TPEF can be set the same as SGA. In Table 4, it shows the results of the number of community K determined by the algorithms. The identified number K is getting far from the real number K when the community structure is more indistinct. The community structure can be identified correctly by using locus-based gene representation methods (i.e. LGA and TPEF) when $u = 0.01$~0.1.

Second, in the community detection analysis, SGA is used as the baseline for the case of number of community K is known,

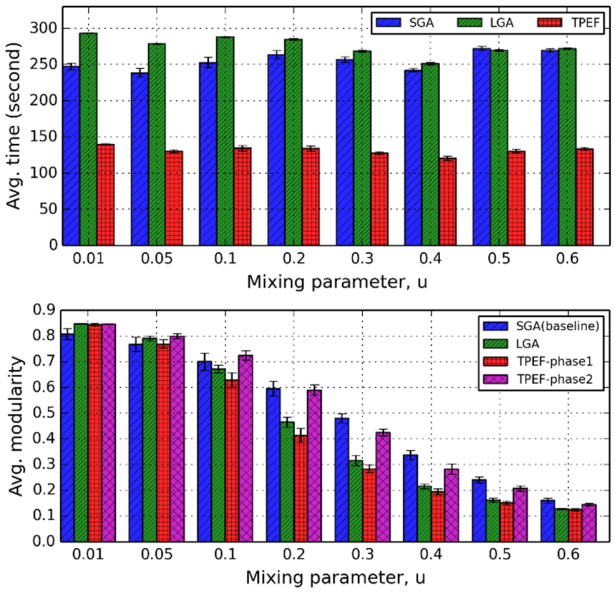

Fig. 2. The execution time and modularity of LFR networks; in the bottom subfigure, the "baseline" means the number of community K of SGA is assigned in advance.

and LGA and TPEF are used for the unknown cases. Therefore, we can observe the difference between TPEF and SGA or LGA in different community structures. The NMI is used to evaluate the quality of identified community structure. In the Fig. 3, TPEF can keep the best NMI value when $u = 0.01$~0.1, and has good NMI value close to SGA and better than LGA when $u = 0.2$~0.6. Hence, TPEF can identify good community structure with the satisfied NMI value when the number of community K is unknown.

Final, in the network spreading ability analysis, the network-based spreading simulation and the SIR model are used to evaluate the spreading ability of multiple network spreaders selected by different measurements and community-based method, as in (6). The simulation is performed 5000 rounds, and each round runs 50 time steps. The $\rho(t)$ is average of cumulative incidence of contagion which means the number of recovered nodes at time t. In the SIR model, the infection rate $rate_{inf} = 0.08$ and recovery rate is $rate_{rec} = 1$. In the selection strategy of multiple network spreaders, the measurements (e.g. betweenness, closeness, degree, k-shell and PageRank) are using the naive top-K strategy which means the top-K nodes are selected as the spreaders; the community-based method is selecting the represented centers of communities as the spreaders.

The Fig. 5 is the network spreading results, using community-based method can infect more nodes than other measurements in a network when $u = 0.01$~0.1. In the Fig. 4, using community-based method to select multiple network spreaders can gain the about average 16% benefit of community structure effect in network spreading when $u = 0.01$; then the benefit is decreasing when u is increasing. On the other hand, when $u \geq 0.2$, using the naive top-K strategy can not only get good network spreading results similar to community-based method, but also save the time of applying the community detection approach.

TABLE III. THE PARAMETERS OF EVOLUTIONARY ALGORITHMS.

Method		num_{evo}	num_{gen}	num_{pop}	$rate_{sel}$	$rate_{cro}$	$rate_{mut}$	K
SGA		20	100	50	0.1	0.8	0.05	Known
LGA								Unknown
TPEF	*Phase1*	20	50	25	0.1	0.8	0.05	Unknown
	Phase2		50	25				

TABLE IV. THE IDENTIFIED NUMBER OF COMMUNITY OF EVOLUTIONARY ALGORITHMS.

Model		LFR model							
Mixing parameter		0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6
Method	SGA	9	8	8	8	9	8	10	9
	LGA	9	8	8	9	10	11	12	20
	TPEF	9	8	8	10	15	10	12	12

V. CONCLUSION AND FUTURE WORK

The community detection approach can be used as a mean of selecting multiple network spreaders in the network. In this study, we assume that the number of community K is unknown, and propose a two-phase evolutionary framework (TPEF) as a trade-off solution between the performance efficiency and the quality of community structure. In the phase one, automatically determine the number of community K according to the topology structure of a network. In the phase two, maximize the modularity of identified community structure produced by the phase one. Final, the multiple network spreaders are selected as the represented centers of communities based on the identified community structure of TPEF.

In the experiment results, the LFR model is used to generate the synthesized networks with different property of community structure (i.e. different mixing parameter u), and to test the proposed algorithm. In the execution time analysis, TPEF can perform well and result the satisfied quality of community structure which is close to the case of number of community is known (i.e. the SGA case). In the community detection analysis, TPEF can hold best NMI value of identified community structure when $u = 0.01{\sim}0.1$, and has good NMI value when $u = 0.2{\sim}06$. In the simulation of network spreading, the measurements (e.g. degree centrality, etc.) and community-based method are compared to analyze the community structure effect in network spreading. In the analysis results, using the community detection approach can gain about average 16% benefit of community structure in network spreading when $u = 0.01$.

The experimental results of this study further imply that the community structure of a network can affect the choice of methods when selecting multiple network spreaders. If the community structure is obvious, i.e. $u = 0.01{\sim}0.1$, the community detection approaches can be used to assist selecting the multiple network spreaders, and gain the benefit of community structure in network spreading. On the contrary, if community structure is unobvious, i.e. $u \geq 0.2$, the naïve top-K strategy based on the well-known measurements is sufficient to select the multiple network spreaders.

As a result, our study points out a potential and interesting research issue: whether the community detection approaches should be applied to assist the selection of multiple network spreaders when the number of community K of a given network is unknown; because the community detection approaches take time but gain much less benefit of community structure in network spreading when applying to a network with a high mixing parameter u. Therefore, in the future work, developing an index, a mechanism or a sampling technic for determining the mixing parameter u is necessary to decide whether applying the community detection approaches to select multiple network spreaders.

REFERENCES

[1] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir, "A model of Internet topology using k-shell decomposition," Proceedings of the National Academy of Sciences, vol. 104, no. 27, pp. 11150–11154, 2007.

[2] B. Doerr, M. Fouz, and T. Friedrich, "Why Rumors Spread So Quickly in Social Networks," Commun. ACM, vol. 55, no. 6, pp. 70–75, Jun. 2012.

[3] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," Nat Phys, vol. 6, no. 11, pp. 888–893, Nov. 2010.

[4] S. Pei and H. A. Makse, "Spreading dynamics in complex networks," J. Stat. Mech., vol. 2013, no. 12, p. P12002, Dec. 2013.

[5] S. Pei, L. Muchnik, J. J. S. Andrade, Z. Zheng, and H. A. Makse, "Searching for superspreaders of information in real-world social media," Sci. Rep., vol. 4, Jul. 2014.

[6] N. A. Christakis and J. H. Fowler, "Social Network Sensors for Early Detection of Contagious Outbreaks," PLoS ONE, vol. 5, no. 9, p. e12948, Sep. 2010.

[7] Y.-H. Fu, C.-Y. Huang, and C.-T. Sun, "Using global diversity and local features to identify influential social network spreaders," in 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2014, pp. 948–953.

[8] M. Newman, Networks: an introduction. Oxford University Press, 2010.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web. 1999.

[10] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," Proceedings of the National Academy of Sciences, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[11] M. E. J. Newman, "Modularity and community structure in networks," Proceedings of the National Academy of Sciences, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[12] M. E. J. Newman, "Communities, modules and large-scale structure in networks," Nat Phys, vol. 8, no. 1, pp. 25–31, Jan. 2012.

[13] M. Rosvall and C. T. Bergstrom, "An information-theoretic framework for resolving community structure in complex networks," Proceedings of the National Academy of Sciences, vol. 104, no. 18, pp. 7327–7331, 2007.

[14] T. Teitelbaum, P. Balenzuela, P. Cano, and J. M. Buldú, "Community structures and role detection in music networks," Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 18, no. 4, p. 043105, 2008.

[15] X. Zhang, J. Zhu, Q. Wang, and H. Zhao, "Identifying influential nodes in complex networks with community structure," Knowledge-Based Systems, vol. 42, pp. 74–84, Apr. 2013.

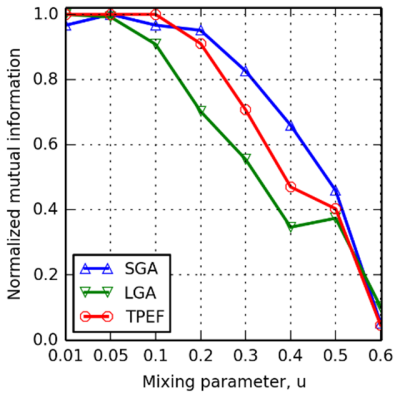

Fig. 3. Normalized mutual information of evolutionary algorithms.

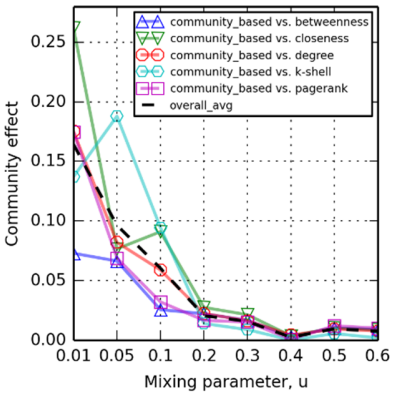

Fig. 4. The benefit of applying the community detection approach.

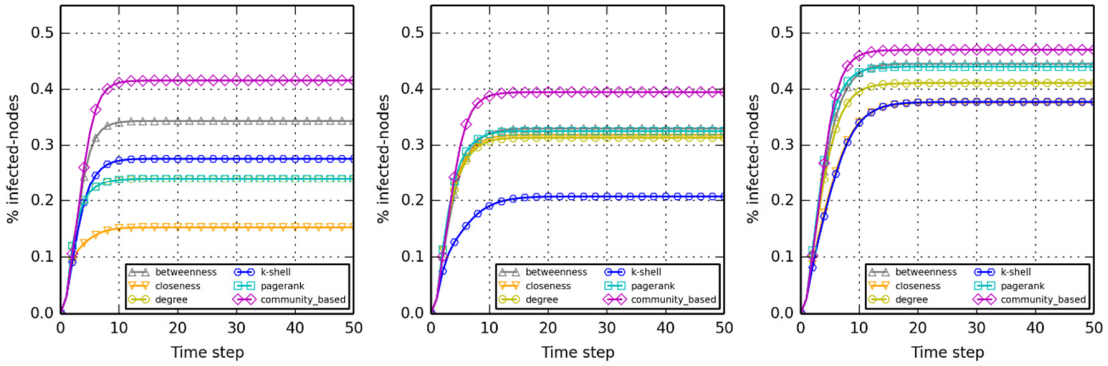

Fig. 5. Network spreading results of multiple network spreaders in LFR networks with u=0.01, 0.05 and 0.1.

[16] S. P. Borgatti, "Identifying sets of key players in a social network," Comput Math Organiz Theor, vol. 12, no. 1, pp. 21–34, Apr. 2006.

[17] T. H. Cormen, Introduction to Algorithms, 3rd Edition. MIT Press, 2009.

[18] C.-Y. Huang and T.-H. Wen, "Optimal Installation Locations for Automated External Defibrillators in Taipei 7-Eleven Stores: Using GIS and a Genetic Algorithm with a New Stirring Operator," Computational and Mathematical Methods in Medicine, vol. 2014, p. e241435, Jun. 2014.

[19] Y.-S. Tsai, P. C.-I. Ko, C.-Y. Huang, and T.-H. Wen, "Optimizing locations for the installation of automated external defibrillators (AEDs) in urban public streets through the use of spatial and temporal weighting schemes," Applied Geography, vol. 35, no. 1–2, pp. 394–404, Nov. 2012.

[20] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," Phys. Rev. E, vol. 78, no. 4, p. 046110, Oct. 2008.

[21] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," Phys. Rev. E, vol. 80, no. 5, p. 056117, Nov. 2009

[22] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach. Prentice Hall, 2010.

[23] N. Matake, T. Hiroyasu, M. Miki, and T. Senda, "Multiobjective Clustering with Automatic K-determination for Large-scale Data," in Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, New York, NY, USA, 2007, pp. 861–868.

[24] T. He and K. C. C. Chan, "Evolutionary community detection in social networks," in 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1496–1503.

[25] R. Shang, J. Bai, L. Jiao, and C. Jin, "Community detection based on modularity and an improved genetic algorithm," Physica A: Statistical Mechanics and its Applications, vol. 392, no. 5, pp. 1215–1231, Mar. 2013.

[26] M. Tasgin, A. Herdagdelen, and H. Bingol, "Community Detection in Complex Networks Using Genetic Algorithms," arXiv:0711.0491 [physics], Nov. 2007.

[27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[28] J. Handl and J. Knowles, "An Evolutionary Approach to Multiobjective Clustering," IEEE Transactions on Evolutionary Computation, vol. 11, no. 1, pp. 56–76, Feb. 2007.

[29] J. Handl and J. Knowles, "Multiobjective clustering with automatic determination of the number of clusters," Technical Report, UMIST, Department of Chemistry, no. TR-COMPSYSBIO-2004–02, Aug. 2004.

[30] C. Pizzuti, "A Multi-objective Genetic Algorithm for Community Detection in Networks," in 21st International Conference on Tools with Artificial Intelligence, 2009. ICTAI '09, 2009, pp. 379–386.

[31] C. Pizzuti, "GA-Net: A Genetic Algorithm for Community Detection in Social Networks," in Parallel Problem Solving from Nature – PPSN X, G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, Eds. Springer Berlin Heidelberg, 2008, pp. 1081–1090.

[32] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," J. Stat. Mech., vol. 2005, no. 09, p. P09008, Sep. 2005.

[33] C. Castellano and R. Pastor-Satorras, "Thresholds for Epidemic Spreading in Networks," Phys. Rev. Lett., vol. 105, no. 21, p. 218701, Nov. 2010.

[34] C.-Y. Huang, C.-T. Sun, C.-Y. Cheng, and Y.-S. Tsai, "Resource limitations, transmission costs and critical thresholds in scale-free networks," in Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2, 2008, pp. 1121–1128.

[35] R. Pastor-Satorras and A. Vespignani, "Epidemic dynamics and endemic states in complex networks," Phys. Rev. E, vol. 63, no. 6, p. 066117, May 2001.