An improved memetic algorithm for community detection in complex networks

Conference Paper · June 2012

DOI: 10.1109/CEC.2012.6252971

CITATIONS
68

READS
803

4 authors, including:

Qing Cai
Nanyang Technological University
59 PUBLICATIONS **2,055** CITATIONS
SEE PROFILE

Li Yangyang
Xidian University
135 PUBLICATIONS **3,067** CITATIONS
SEE PROFILE

An Improved Memetic Algorithm for Community Detection in Complex Networks

Maoguo Gong, Qing Cai, Yangyang Li, Jingjing Ma
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education
Xidian University, Xi'an, China

***Abstract*—There is an increasing recognition on community detection in complex networks in recent years. In this study, we improve a recently proposed memetic algorithm for community detection in networks. By introducing a Population Generation via Label Propagation (PGLP) tactic, an Elitism Strategy (ES) and an Improved Simulated Annealing Combined Local Search (ISACLS) strategy, the improved memetic algorithm called (iMeme-Net) is put forward for solving community detection problems. Experiments on both computer-generated and real-world networks show the effectiveness and the multi-resolution ability of the proposed method.**

Keywords-community detection; memetic algorithm; label propagation; elitism strategy; simulated annealing

I. INTRODUCTION

In reality, many complex systems can be represented as networks. Complex collaboration networks [1], the world-wide-web [2, 3], biological networks such as neural networks [4], food webs [5], and metabolic networks [6, 7], and social networks [8, 9], are just some good examples. Complex networks contain a lot of efficacious information that can be used to mining the relationships among objects that composing many real world systems. Networks are commonly modelled as graphs, where nodes represent the objects and edges represent the interactions amongst these objects. One of the most important and also challenging task in complex networks analysis is the detection of community structure. Though there is not a standard definition for community, in literatures, it is universally acknowledged that communities are some sub-graphs whose intra-connection is dense, while on the contrary, the inter-connection is rather sparse. During the past several decades, a lot of scholars had been doing research on analyzing the community structure in complex networks. So a large amount of algorithms have been proposed [10-22].

To some extent, the detection of community structure in a network can be considered as a problem of clustering, and thus, it can be formally defined as an optimization problem [23]. Obviously this implies the choice of an appropriate objective function determines the clustering performance. In the last few years, based on this idea, many different approaches have been proposed to uncover the hidden community structure behind networks [24-30] (a recent review can be found in [31]). All these approaches define the criterion functions and try to find the clustering method that best optimizes them. In particular, Girvan and Newman [32] used the concept of modularity as a criterion to stop the division of a network into sub-networks in their divisive hierarchical clustering algorithm (GN). GN is one of the most known community detection methods.

The motivation for modularity maximization is based on the following assumption: high modularity values indicate the best partition or at least a very good one for a given graph. Therefore many algorithms have been employed to optimize modularity, to name a few, greedy algorithm [23, 25], simulated annealing [33], extremal optimization [34], etc. In a work done by Tasgin et al. [30], they chose modularity as the fitness function and implement genetic algorithm to find the best partition of networks. The algorithm does not require prior knowledge such as the real clusters of the section. In another work accomplished by Liu et al. [35], the maximum modularity partition is obtained via iterative bi-partitions of the graph, where each bi-partition is determined by applying a genetic algorithm to each subgraph.

However, Fortunato and Barthélemy have pointed out in [36] that there exist the resolution limitation in modularity optimization. That is to say, algorithms based on modularity may fail to identify modules smaller than a scale which depends on the total size of the network and on the degree of inter connectedness of the modules, even in the case scenario where modules are unambiguously defined. Li et al. [37] have introduced a quality function, called modularity density. The authors have introduced a tunable parameter which allows one to explore the network at different resolutions. It has been proved that modularity density is superior to the widely used modularity.

Gong et al. [38] have introduced a Memetic Algorithms (MAs). The method is a combination of Evolutionary Algorithms (EAs) with local search. MAs are inspired by the concept of a meme first proposed by Richard Dawkin. Meme represents a unit of cultural evolution that can exhibit local refinement [39]. MAs are also called Hybrid Genetic Algorithms, Genetic Local Searchers, Lamarckian Genetic Algorithms, etc. From the angle of optimization, MAs have been demonstrated to be more efficient and more effective than traditional EAs for some problem domains, especially in combinational optimization field [40, 41]. Local search is a means for algorithms to find global optimal solutions rather than local ones. The widely used local search means are Hill Climbing (HC) [42], Simulated Annealing (SA) [43-46] and Taboo Search (TS) [47-49]. In this paper, an improved SA approach is applied to substitute the HC operator, because HC is easy to fall into local optimal situation, while SA just can overcome this drawback. Besides, in order to speed up the

This work was supported by the Program for New Century Excellent Talents in University (Grant No. NCET-08-0811), the Program for New Scientific and Technological Star of Shaanxi Province (Grant No. 2010KJXX-03), and the Fundamental Research Funds for the Central Universities (Grant No. K50510020001).

convergence speed, here another two tactics are implemented. One is named Label Propagation (LP), the other is Elitism Strategy (ES). Thus, an improved memetic algorithm for community detection in complex networks is proposed. In the following, we denote the algorithm as iMeme-Net. Experiments on computer-generated and real-world networks show that the proposed algorithm is more efficient and much faster than that of Meme-Net. By tuning the parameter lamada in the quality function, we are able to explore the network at different resolutions.

II. PRELIMINARY

A. Modularity Density

Modularity density is firstly proposed by Li et al. [37]. In order to give a brief introduction of modularity density, first let us consider an undirected graph $G=(V,E)$, where V is the vertex set and E is the edge set. The adjacent matrix of the graph is A whose element $a_{ij}=1$, if node i and node j has a connection, otherwise 0. We define a partition of a network as: $\Omega=\{s_1,s_2,...,s_m\}$, where $1\le m\le n$, n is the number of nodes and m is the number of communities. Say $c_1,c_2\in\Omega$, we further define $L(c_1,c_2)=\sum_{i\in c_1,j\in c_2}a_{ij}$, $L(c_1,c_1)=\sum_{i\in c_1,j\in c_1}a_{ij}$, $L(c_1,\overline{c_1})=\sum_{i\in c_1,j\in \overline{c_1}}a_{ij}$, where $\overline{c_1}=\Omega-c_1$. Then the modularity density can be written as:

$$D=\sum_{i=1}^{m}\frac{L(c_i,c_i)-L(c_i,\overline{c_i})}{|c_i|}. \tag{1}$$

where $|c_i|$ is the number of nodes in cluster c_i.

In the equation, each summand is the ratio between the difference of the internal and external degrees of the subgraph G_i and the size of the subgraph. Normally we take it that the larger the value of D, the more accurate the partition is.

The authors also have proved the equivalence form of modularity density and kernel k means, and therefore give a more general form of D which can be calculated as follows:

$$D_\alpha=\sum_{i=1}^{m}\frac{2\alpha L(c_i,c_i)-2(1-\alpha)L(c_i,\overline{c_i})}{|c_i|}. \tag{2}$$

where α is a resolution controlling parameter. When $\alpha=0$, D_α is equivalent to the ratio cut [50]; when $\alpha=0.5$, D_α is equal to the modularity density D defined in (1); when $\alpha=1$, D_α is equivalent to the ratio association [50]. As is pointed out in [51] that optimization of the ratio association often clusters a network into small parts, while the ratio cut the large communities. To optimize the D_α value by interpolating the α parameter, we can get modules of a network with different resolutions.

B. Label Propagation

The idea of label propagation through a network has been studied by Bagrow [52] in his L-shell method. Label propagation is like the spreading of a kind of vicious epidemic disease. Suppose person A is infected with the disease, then he spreads the disease to one of his friends, say person C, let us say C has friend B, but B and A do not know each other (in the network, it corresponds to the situation that there is not an edge between node A and B), however, C transmits the virus to B, then, though A and B have not directly gotten in touch with each other, all of them are infected, and they should be isolated (similarly, in the network, node A, B and C are separated into the same community).

The further illustration of label propagation is the following. Suppose that a node i has neighbour set $\Omega(i)=(x_1,x_2,...,x_k)$ and let $l(i)$ be the community label to which node i belongs to. At first we initialize every node with unique labels, i.e., $l(i)=i$. Then each node determines its community label based on the labels of its neighbors. Next step we let the labels propagate through the network. We assume that each node in the network chooses to join the community to which the maximum number of its neighbors belong to, and this can be represented as the following formula:

$$l(i)=\arg\max_r\sum_{j\in\Omega(i)}\delta(l(j),r). \tag{3}$$

where $\delta(i,j)=1$, if node i and j belong to the same community, otherwise 0. As the labels propagate, densely connected groups of nodes quickly reach a consensus on a unique label. We perform this process iteratively, where at every step, each node updates its label based on the labels of its neighbours. As is pointed out in [53] that when the propagation iterations reaches five, and if we form all the community labels to be a chromosome, then the created candidate individual possesses high clustering accuracy, and besides, we can obtain considerable diversity of the population.

Similar idea is studied by Costa in [54]. In literature [55], Wu proposed a novel method. The network is regarded as an electric circuit, and a battery is attached to two random nodes that are supposed to be within two communities. The algorithm partitions a network into two communities. Although this method can be generalized to detecting multiple communities, it requires the number of communities as the input, what is more, it tends to discover communities of approximately the same size.

C. Simulated Annealing

SA is a generic probabilistic metaheuristic. It is widely applied for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. It is often used when the criteria space is discrete. For certain problems, SA may be more suitable and efficient than exhaustive enumeration or simple greedy approach — provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible one.

The name and inspiration of SA come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and

reduce their defects. The heat causes the atoms within the material to become unstuck from their initial state and wander randomly through levels of higher energy, while the cooling process gives them more chances of finding configurations with lower internal energy than the initial one.

By analogy with this physical process, each step of the SA algorithm attempts to replace the current solution by a new solution (normally the new solution is acquired by applying a kind of transformation to the current solution). Provided that the corresponding function values of the new solution is better than that of the current ones, then we accept the new solution, or the new solution may then be accepted with a probability that depends both on the difference between the corresponding function values and also on a global parameter *T* (called the temperature), that is gradually decreased during the process. Compared with exhaustive enumeration and simple greedy approaches, SA saves the method from being stuck at local optima—which are the bane of greedier methods.

The method was independently described by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi in 1983 [43], and by Vlado Černý in 1985 [44]. It is an adaptation of the Metropolis-Hastings algorithm, a Monte Carlo method to generate sample states of a thermodynamic system, invented by M.N. Rosenbluth in a paper by N. Metropolis et al. in 1953 [45]. And a proof of convergence of SA can be found in [46].

D. Elitism Strategy (ES)

ES is a kind of selection tactic for preserving the best individual among the offspring individuals, at the same time, ES can improve the convergence speed. Say variable *bestind* is occupied to store the best individual, if the best individual from the current population is better than *bestind*, then copy its traits to *bestind*, else, assign *bestind*'s properties to the worst individual in the current population. From the view point of evolution, since best genes in the previous population substitute the worst ones, with the evolution going on, ideal individual will appear after several times' genetic operations. In order to obtain a tradeoff between accuracy and speed, we only implement one elite individual in our proposed algorithm.

III. ALGORITHM DESCRIPTION

In this section, we will give a detailed description about the proposed algorithm: iMeme-Net. As is discussed above, our goal is to maximize the modularity density function in (2). After taking a deep reconsideration of and a thoroughly experimental testing of the algorithm framework, the final framework of iMeme-Net adopted in this paper is given as *Algorithm 1*:

Next, a further explanation of the algorithm is giving as follows:

A. Population Generation via Label Propagation (PGLP)

In this paper, the string based coding is utilized. Thus, a chromosome can be denote as:

$$chrom(i) = \{l_1, l_2, ..., l_n\}. \quad (4)$$

where n is number of nodes of the graph, l_i is the label of node i and it is a random integer between 1 and n. If l_i equals to l_j, then we say that node i and j are in the same community.

Apparently, this representation does not need the number of clusters present in the graph, which actually is a result of our algorithm. A graph of n vertices can be partitioned into n clusters at most. If we initialize population like (4) , then the population is lack of diversity and each solution is of low quality. In [38] the authors initialize the population in the following way: first, assign a unique label to each node, that is to say: $chrom(i) = \{1,2,...,n\}$, next, for each chromosome, randomly select a vertex and assign its label identifier to all of its neighbors (vertices that have edges connected with the selected vertex), and this operation is repeated αn times for each chromosome, where α is a parameter and $\alpha = 0.2$ is used in that paper. Inspired by [53] we proposed another approach which is proved to be more efficient and can speed up the convergence of the whole algorithm. The population is generated via the label propagation approach mentioned above. The procedure is: firstly assign a unique label to each node, secondly, for every vertex, implement formula (3) to update labels of each node. For each chromosome, this step is executed *k* times where *k* is the propagation iteration number. Experiment shows that PGLP is superior to the one depicted in [38]. The pseudo code is giving in *Algorithm 2*.

Algorithm 1 Framework of iMeme-Net

Begin
Input: Adjacency matrix *A* of the graph.
parameters: *popsize*, *gens*, *pm*, *temperature*.
1: **pop** ← PGLP(*popsize*);
2: **pop** ← EvaluateFitness(**pop**);
3: **bestpop** ← KeepBestIndividual(**pop**);
4: **for**(i=1:*gens*)
NewPop ← NeighborBasedMutate(**pop**);
TempPop1 ← ISACLocalSearch(**NewPop**);
TempPop2 ← ElitismPreservation(**TempPop1**);
Pop ← UpdatePopulation(**TempPop1** & **TempPop2**);
End for
Output: *BestDensity*, *partitions*.
End

Algorithm 2 Pseudo Code of PGLP

```
Begin
Input: label propagation iterations: iters.
    for each chrom(i)∈ population
      for(j = 1 : iters)
        for(k = 1 : vertexes)
          if(node[k].neighbor.size > 1)
            for(m = 1 : node[k].neighbor.size)
              Node[k].label ← formula (3);
          else Node[k].label ← node[k].neighbor.label;
        End for
      End for
    End for
Output: initialed chromosomes.
End
```

B. Neighbour Based Mutation (NBM)

For the genetic operators, we only retain the mutation operation. Because crossover operation will destroy good gene blocks built by the very initialization procedure, especially when the network contains triangles or circles. The procedure can be depicted as follows: first we generate a pseudo random number, for each chromosome, if the random number is smaller than the mutation probability, the NBM process is applied to the chromosome, namely, randomly chose one vertex, then change its label identifier to one of its neighbours'. This operation is repeated *n* times on the chromosome where *n* is the nodes number.

C. Improved Simulated Annealing Combined Local Search (ISACLS) procedure

Originated from the technique of annealing in metallurgy, in this paper, a ISACLS tactic is proposed. However, there is a subtle difference. According to the Metropolis principle, the possibility P for a particle to reach the balanced state at temperature T can be depicted by the following formula:

$$P = \exp(^{-|\Delta E|}/_{rT}). \quad (5)$$

where r is the Boltzmann constant, T is the original temperature, and ΔE is the internal energy difference. The original SA is a iterative process, because after each iteration, the final temperature is changed, i.e., $t = r * T$, the iteration will not stop until t reaches a small value, usually we set $T = 500$, $r = 0.9$, and the stop criteria is $t \geq 0.5$, thus, the SA process takes a quite long time, especially when the scale of the network is large. In order to save time and simulate the SA process, in this paper, we proposed a new version for the very community detection problems which is written as follows:

$$P = \exp(^{-|f_c - f_m|}/_{\beta}). \quad (6)$$

Where f_c and f_m are the modularity density values of the current individual's and the individual after been mutated, respectively. The β parameter is a constant, through experiment test, we set its value as 0.16. In order to give a detailed description of the ISACLS approach, we first define the partition of the graph and the neighbors of the partition. Given a partition $\Omega = \{s_1, s_2, ..., s_m\} (2 \leq m \leq n)$, where m is the number of clusters of this partition and n is the number of vertices in the graph. Now we randomly choose a single vertex *p*, say it in cluster s_i, and then, we delete node *p* in cluster s_i and reassign it to cluster $s_j (i \neq j)$. Therefore, several new partitions $\Omega^{'}$ is obtained, and we call it the neighbors of the partition Ω. By the way, we consider a special situation that when $m = 1$, $\Omega = \{s_1\}$, a neighbor of Ω is defined as $\Omega^{'} = \{\{s_1 - p\}, \{p\}\}$. Now having these definitions, the main idea of SACLS can be illustrated as: we randomly choose a chromosome, decode it to a section Ω, and then calculate its neighbors section $\Omega^{'}$, figure out the fitness f of Ω and $f^{'}s$ of $\Omega^{'}$, and then find out the best one among $f^{'}s$, and we record it as $f_{c\max}$, if $f_{c\max}$ is superior to f, then we copy the corresponding genes to the selected chromosome, or we accept this solution with the probability *p* which is worked out through (6). Since we retain bad solutions with certain probability and let them keep on evolving, the algorithm will step out local optimal and eventually reaches global optimal solution. This point has been proved by our later experiments. The pseudo code is giving in *Algorithm 3*.

Algorithm 3 Framework of ISACLS

Begin

Input: constant β, seeds of system clock, *popsize*.

1 $chrom \leftarrow rand(chrom(popsize))$;
2 $\Omega \leftarrow decode(chrom)$;
3 $\Omega^{'} \leftarrow CalculateNeighbors(\Omega)$;
4 $f^{'}s \leftarrow fitness(\Omega^{'}), f \leftarrow fitness(\Omega)$;
5 $f_{c\max} = \max(f^{'}s)$;
6 **if** ($f < f_{c\max}$)
7 $chrom \leftarrow chrom(\Omega^{'}_{c\max})$;
8 **else if** ($rand(1) < \exp(^{-|f - f_{c\max}|}/_{\beta})$)
9 $chrom \leftarrow chrom(\Omega^{'}_{c\max})$;

Output: *chromosome that has been implemented local search operation*.

End

IV. Experimental Results

This part, we apply our algorithm to both computer-generated networks and real-world networks.

A. Experimental Settings

All the algorithms are written in C++, the experiments have been performed on a Inter(R) Core(TM) i3 machine, 3.19GHz, 3.05 GB memory. All the experimental parameters are listed in TABLE I, where *iter* is the label propagation times, *pop* represents the population size and *maxgene* denotes the population evolution iterations.

B. Evaluation Metrics

Checking the performance of an algorithm usually involv-

TABLE I. Experimental parameters

	pop	maxgene	tour	β	iter	pool	pc	pm
iMeme-Net	100	50	—	0.16	5	—	—	0.9
Meme-Net	450	50	2	—	—	225	0.8	0.2

es defining criterions to establish how similar the partition delivered by the algorithm is to the expected one. In literatures, the quality of a partition can be evaluated by using validity indices. The validity indices can be both internal, i.e. they rely on the connections and separation between the groups, and external, i.e. using additional data to assess the clustering performance. For the case when the real communities of a network are known, we adopt an external measure, called *Normalized Mutual Information* (*NMI*), described in [55], to estimate the similarity between the true partitions and the detected ones.

Given two partitions *A* and *B* of a network in communities, let *C* be the confusion matrix whose element C_{ij} is the number of nodes shared in common by community *i* in partition *A* and community *j* in partition *B*. The $NMI(A,B)$ is then defined as:

$$NMI = \frac{-2\sum_{i=1}^{CA}\sum_{j=1}^{CB} C_{ij}\log(C_{ij}N / C_{i.}C_{.j})}{\sum_{i=1}^{CA} C_{i.}\log(C_{i}/N) + \sum_{j=1}^{CB} C_{.j}\log(C_{.j}/N)}. \quad (7)$$

where $CA(CB)$ is the number of clusters in partition *A*(*B*), $C_{i.}(C_{.j})$ is the sum of elements of C in row *i* (column *j*), and N is the number of nodes. If *A* = *B*, then $NMI(A,B)=1$; if *A* and *B* are totally different, then $NMI(A,B)=0$. The *NMI* is a similarity measure proved to be reliable by Danon et al. [56].

C. Results on Synthetic Networks

Testing an algorithm essentially means applying it to a specific problem whose solution is known and comparing such solution with that delivered by the algorithm, thus, we first do some experiments on the benchmark networks proposed by Lancichinetti et al. [57], which is an extension of the classical benchmark proposed by Girvan and Newman in [32]. The benchmark network consists of 128 nodes divided into four communities of 32 nodes each. Every node has an average degree of 16 and shares a fraction γ of links with the other nodes of its community, and $1-\gamma$ with the other nodes of the network. Here, γ is called the mixing parameter. When $\gamma < 0.5$, the neighbours of a vertex inside its community are more than the neighbors belonging to the rest groups, thus a good algorithm should discover them. We test iMeme-Net and Meme-Net on 10 computer-generated networks with the values of γ ranging from 0.1 to 0.5, and the values of λ ranging from 0.2 to 1.0. We use the *NMI* metric to measure the similarity between the true partitions and the detected ones.

Fig.1 and Fig.2 display the *NMI* values averaged over 10 runs for different values of the resolution controlling parameter λ when the mixing parameter γ increases from 0.0 to 0.5 with interval 0.05.

As is shown in Fig.1 and Fig.2, when the mixing parameter γ is no bigger than 0.4, both methods can figure out the true partitions (NMI equals 1).As the mixing parameter increases, the community structure in the network is becoming fuzzy gradually, it is more and more difficult to detect the true

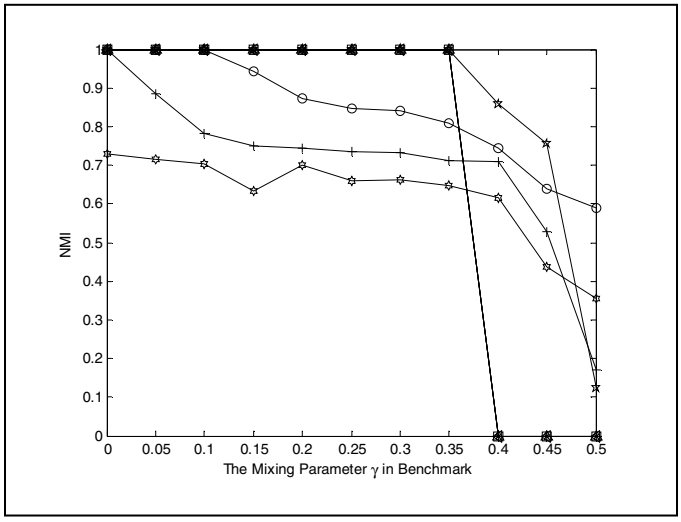

Figure 1. Benchmark results obtained by Meme-net.

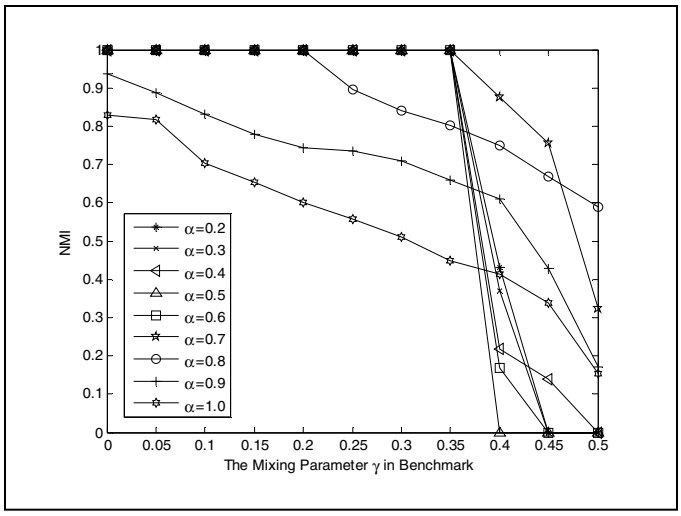

Figure 2. Benchmark results obtained by iMeme-net.

partitions. By alerting the value of parameter α, this phenomenon can be improved. As is illustrated in the figure, larger α can help to identify more vague community structure than small α does. For example, when α is smaller than 0.7 and γ is larger than 0.4, both algorithms regard the whole network as one group (NMI equals 0). However, when α reaches 0.8 and say γ is 0.45, we get 7 clusters and the corresponding NMI is 0.668. Notice that, when $\gamma = 0.5$, since each node has half of the links inside the community and the other half with the rest of the network, the community structure is rather fuzzy, in this case, hardly any algorithm can find the true partition of the network. For example, when $\gamma = 0.5$ and α, we discovered 9 communities, the NMI is quite low and the section is rather bad. In general, both algorithms yield the similar performance on synthetic data sets.

D. Results on Real-World Networks

We now show the applications of iMeme-Net on four real-world networks whose structures are known: the Zachary's

karate club, the Dolphin social network, the American College football, and the Books about US politics and three real-world networks whose structures are known: Power Grid, High-Energy Theory Collaborations, Astrophysics collaborations.

Zachary's karate club: The Zachary's Karate Club network (34 vertices, 78 edges) was complied by Zachary. He observed 34 members of a karate club over a period of two years [58]. Because of fierce dispute developed between the administrator of the club and the club's instructor, the instructor left with some members and found up a new club. Ultimately, the club is separated into two groups.

Dolphin social network: By observing 62 bottlenose dolphins' behavior during seven years' living in Doubtful Sound, New Zealand, Lusseau constructed the network [59] (62 vertices, 160 edges). A tie between two dolphins was established by their statistically significant frequent association. The network naturally is separated into two large groups, the female group and the male one.

American College football: This network (115 vertices, 613 edges) was established by M. Girvan and M. Newman [32]. It represents American football games between Division IA colleges during regular season Fall 2000. Nodes in the graph represent teams (identified by their college names) and edges represent the games between the two teams they connect. The network is grouped in 12 communities.

Books about US politics: This is a network (105 vertices, 441 edges) of books on politics, founded by V. Krebs and is used for demonstration in [60]. The nodes represent books on American politics bought from Amazon.com while edges between books represent frequent co-purchasing of books by the same buyers. By reading of the descriptions and reviews of the books posted on Amazon, the books are eventually classified into 3 categories.

The range of the resolution controlling parameter α used in our experiments is 0.2 to 1 with interval 0.1. For each network, we ran our algorithm 10 times and recoded the best results. The results of the proposed algorithm are reported in tables from II to V. The data in brackets are the results yielded by Meme-Net which are used as an comparison of our algorithm.

On Zachary's Karate Club network, both algorithm get the same results. For $\alpha = 0.3$, both algorithms can detect the true communities in the network which correspond to NMI = 1. For $\alpha = 0.4$, it separates the network into 3 modules which is meaningful. By observation, we find that it abstracts nodes 5, 6, 7, 11, 17 from the first community and forms a new one. This is similar to the case when $\alpha = 0.5$. For α is 0.6 (0.7, 0.8), the corresponding communities are the sub-division of the original ones. So we can see that larger values of α will lead to smaller cliques, $\alpha = 0.9$ and $\alpha = 1.0$ are just two good examples. However, it never misplaced any vertex. Additionally, with the application of elitism tactic and label propagation strategy, the improved algorithm is much faster than Meme-Net. For the Karate data set, the former one solved the problem within one second while the latter one cost about 10 seconds. The time disparity will be larger when experimenting on larger data set. To further illustrate this point, we undertake a small test on Karate Club data set. We randomly generate two populations (depicted as popA and popB in Fig.3) containing 10 chromosomes each. The popA is generated via label propagation approach while the popB is delivered by the method mentioned in our previous version. After decoding the chromosomes, we just apply the NMI metric to compute the similarities between the obtained partitions and the true ones. From Fig.3 we can see that it is apparent the introduced PGLP tactic is superior to the one utilized in our previous paper. To generate populations via label propagation one can get population individuals with high efficiency. From the very view point of evolutionary, this kind of approach can speed up the convergence of the whole algorithm, cause it saves the criterion searching space.

As for Dolphin social network, for $\alpha = 0.3$, 2 clusters are found and the corresponding NMI is 1, both algorithms can identify the real structure, but when α reaches 0.4, the

TABLE II. RESULT ON KARATE CLUB NETWORK

α	NMI	D(α)	clusters
0.2	0(0)	1.835(1.835)	1(1)
0.3	1(1)	3.156(3.156)	2(2)
0.4	0.699(0.699)	5.254(5.254)	3(3)
0.5	0.699(0.699)	7.845(7.845)	3(3)
0.6	0.687(0.687)	10.972(10.972)	4(4)
0.7	0.687(0.687)	14.438(14.438)	4(4)
0.8	0.687(0.687)	17.902(17.902)	4(4)
0.9	0.443(0.443)	24.571(24.571)	9(9)
1.0	0.455(0.455)	32.952(32.952)	12(12)

TABLE III. RESULT ON DOLPHIN NETWORK

α	NMI	D(α)	clusters
0.2	0.889(0.889)	3.112(3.112)	2(2)
0.3	1(1)	5.103(5.103)	2(2)
0.4	1(0.653)	7.608(7.099)	2(2)
0.5	0.597(0.520)	10.883(9.909)	4(3)
0.6	0.673(0.438)	15.166(12.522)	5(3)
0.7	0.475(0.383)	24.203(22.278)	8(11)
0.8	0.367(0.337)	35.468(33.133)	14(13)
0.9	0.360(0.337)	49.789(48.001)	16(17)
1.0	0.328(0.331)	68.667(51.468)	23(23)

TABLE IV. RESULT ON COLLEGE FOOTBALL NETWORK

α	NMI	D(α)	clusters
0.2	0(0.610)	4.264(-2.577)	1(4)
0.3	0(0.748)	6.397(6.196)	1(6)
0.4	0.547(0.879)	15.643(12.456)	3(11)
0.5	0.756(0.889)	29.321(28.173)	8(14)
0.6	0.863(0.901)	56.242(57.998)	10(13)
0.7	0.878(0.839)	85.921(80.636)	10(10)
0.8	0.862(0.857)	110.256(105.493)	12(12)
0.9	0.886(0.858)	140.495(134.115)	14(13)
1.0	0.885(0.865)	181.571(179.281)	20(17)

TABLE V. RESULT ON POLITICS BOOKS NETWORK

α	NMI	D(α)	clusters
0.2	0.598(0.598)	5.272(5.272)	2(2)
0.3	0.598(0.598)	8.632(8.632)	2(2)
0.4	0.598(0.574)	11.992(13.919)	2(3)
0.5	0.570(0.458)	20.160(20.186)	4(6)
0.6	0.509(0.548)	27.021(25.710)	5(6)
0.7	0.525(0.530)	38.529(36.452)	7(6)
0.8	0.411(0.408)	57.347(53.817)	11(11)
0.9	0.402(0.406)	76.897(63.705)	15(15)
1.0	0.386(0.363)	116.495(115.552)	35(31)

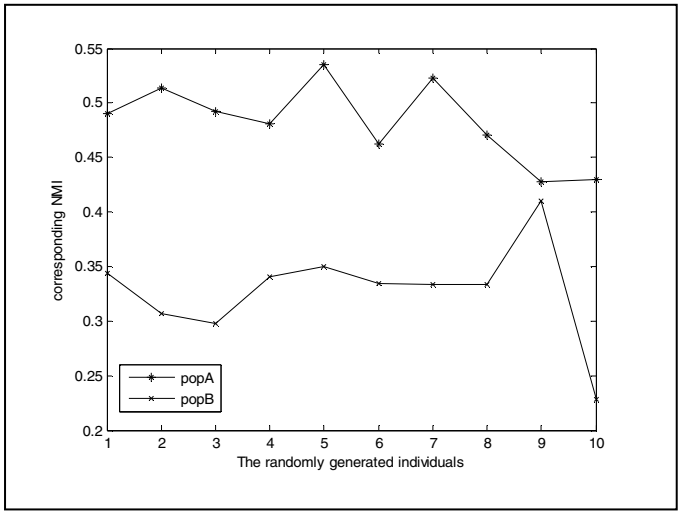

Figure 3. Comparison between two population generation means.

improved version still can find the true classification, though there is a gentle decrease on the objective function. The decrease is caused by the ISACLS procedure since it accepts bad solutions with certain probability, sometimes local optimal results can occur, but it is just a small probability case. As α increases, smaller clusters appears, which are the sub-division of bigger communities, however, never has a vertex been misplaced. For α increasing from 0.5 to 0.9, there are common increases on both the NMI and the modularity density value. It clearly shows that the hill climbing method is easy to fall into local optimal situation while the ISA approach just can overcome this drawback. From the last result, though the objective function's value goes up, the NMI gets down, we can draw the conclusion that larger function value does not equal to bigger NMI index. This point can be proved by the experiments results on the latter networks.

The College Football network is a weighted network. table 3 suggests that the modularity density function maybe is not proper for weighted networks. From the table we can see that the largest NMI obtained by the improved algorithm is 0.886 which separates each one of the two real communities into 2 smaller one, and thus, 14 clusters are found. Though when $\alpha = 0.8$, both methods find 12 groups which is actually the expected numbers of clustering, this partition misplaced several nodes . The iMeme-Net method misplaced vertexes 58, 59, 63, 82, 97 and 110 and Meme-Net wrongly separated nodes 18, 26, 31, 32 and 100.

Experiments on the Politics Books network appear the same phenomenon with that of College Football network. As what is gathered in table 4, for $\alpha = 0.3$, both algorithms group the network into two big parts. The original second and the third community are merged to form a bigger one. For $\alpha = 0.4$, iMeme-Net wrongly partitioned nodes 50, 52, 59, 60, 65, 68, 69, 70 and 105. Meme-Net discovered 3 communities, however, nodes 6, 29, 52, 56, 57, 58, 77 and 60 are displaced. Though the classification results remain much to be expected, we still get a gentle increase on the objective function values.

From the above experimental results we can draw the conclusion that the proposed algorithm is superior to our previous proposed version.

V. CONCLUSION

In this paper, we propose an improved Memetic-Net algorithm which is superior to the previous version. By introducing a Population Generation via Label Propagation tactic, an Elitism strategy and an improved Simulated Annealing approach, our proposed algorithm can yield better performance with smaller population size than our previous version, and its much faster. Experiments on both computer-generated and real-world networks demonstrate the high efficiency and the reliability of the proposed algorithm. By tuning the parameter in the objective function, our algorithm is capable of discovering muti-resolution communities in a network.

References

[1] Newman, M. E. J. (2001) The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences 98 (2): 404–409.

[2] Albert, R., Jeong, H. & Barabási, A.-L. (1999) Diameter of the World-Wide Web. Nature (London) 401, 130–131.

[3] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. & Wiener, J. (2000) Graph structure in the web. Comput. Networks 33, 309–320.

[4] Watts, D. J. & Strogatz, S. H. (1998) Collective dynamics of 'small-world' networks. Nature (London) 393, 440–442.

[5] Williams, R. J. & Martinez, N. D. (2000) Simple rules yield complex food webs. Nature (London) 404, 180–183.

[6] Jeong,H., Tombor, B., Albert, R.,Oltvai, Z. N. & Barabási, A.-L. (2000) The large scale organization of metabolic networks. Nature (London) 407, 651–654.

[7] Fell, D. A. & Wagner, A. (2000) The small world of metabolism. Nat. Biotechnol. 18, 1121–1122.

[8] Wasserman, S. & Faust, K. (1994) Social Network Analysis. Cambridge University Press, Nov 25, page 825.

[9] Scott, J. (2000) Social Network Analysis: A Handbook, 2nd Ed. Newberry Park, CA: Sage. ISBN 0-7619-6338-3

[10] Pool, I. de S. & Kochen, M. (1978) Contacts and influence. Social Networks 1, pp. 1–48.

[11] Milgram, S. (1967) The Small World Problem. Psychol. Today 2, pp. 60–67.

[12] Barabási, A.-L. & Albert, R. (1999) Emergance of scaling in random networks. Science 286, 509–512.

[13] Krapivsky, P. L., Redner, S. & Leyvraz, F. (2000) Connectivity of growing random networks. Phys. Rev. Lett. 85, 4629–4632.

[14] Dorogovtsev, S. N., Mendes, J. F. F. & Samukhin, A. N. (2000) Structure of growing networks: Exact solution of the Barabasi-Albert model. Phys. Rev. Lett. 85, 4633–4636.

[15] Newman, M. E. J., Strogatz, S. H. & Watts, D. J. (2001) Random graphs with arbitrary degree distributions and their applications. Phys. Rev. E 64, 026118.

[16] Redner, S. (1998) How Popular is Your Paper? An Empirical Study of the Citation Distribution. Eur. Phys. J. B 4, 131–134.

[17] Menger, K. (1927) Zur Allgemeinen kurventheorie. Fundamenta Mathematicae 10, 96–115.

[18] White, D. R. & Harary, F. (2001) The cohesiveness of blocks in social networks. Sociol. Methodol. 31, 305–359.

[19] Ahuja, R. K., Magnanti, T. L. & Orlin, J. B. (1993) Network Flows: Theory, Algorithms, and Applications Prentice–Hall, Upper Saddle River, NJ, USA. ISBN 0-13-617549-X.

[20] Katz, L. (1953) A new status index derived from sociometric analysis. Psychometrika 18, 39–43.

[21] Freeman, L. (1977) A set of measures of centrality based on betweenness. Sociometry 40, 35–41.

[22] Newman, M. E. J. (2001) Scientific collaboration networks: I. Network construction and fundamental results. Phys. Rev. E 64, 016131.

[23] Newman, M. E. J. (2004) Fast algorithm for dtecting community structure in network. Phys. Rev. E, 69(6):066133.

[24] Arenas, A. and Diaz-Guilera, A. (2007) Synchronization and modularity in complex networks. European Physical Journal ST, 143:19–25.

[25] Clauset, A., Newman, M. E. J., Moore, C. (2004) Finding community structure in very large networks. Physical Review E 70, 066111.

[26] Lozano, S., Duch, J., and Arenas, A. (2007) Analysis of large social datasets by community detection. European Physical Journal ST, 143:257–259.

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

[28] Clara Pizzuti. (2008) Ga-net: a genetic algorithm for community detection in social networks. In Proc. of the 10th International Conference on Parallel Problem Solving from Nature (PPSN 2008), pages 1081–1090.

[29] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. (2004) Defining and identifying communities in networks. Proc. Natl. Acad. Sci. USA (PNAS'04), 101(9):2658–2663.

[30] Mursel Tasgin, Amac Herdagdelen, and Haluk Bingol. (2007) Communities detection in complex networks using genetic algorithms. oai:arXiv.org:0711.0491v1 [physics.soc-ph].

[31] Santo Fortunato and Claudio Castellano. (2007) Community structure in graphs. arXiv:0712.2716v1 [physics.soc-ph].

[32] Girvan, M. and Newman, M. E. J. (2002) Community structure in social and biological networks. In Proc. National. Academy of Science. USA 99, pages 7821–7826.

[33] Roger Guimerà, Marta Sales-Pardo, and Luís A. Nunes Amaral. (Aug 2004) Modularity from fluctuations in random graphs and complex networks. Phys. Rev. E, 70(2):025101.

[34] Jordi Duch and Alex Arenas. (Aug 2005) Community detection in complex networks using extremal optimization. Phys. Rev. E, 72(2):027104.

[35] Xin Liu, Deyi Li, Shuliang Wang, and Zhiwei Tao. (2007) Effective algorithm for detecting community structure in complex networks based on ga and clustering. In Computational Science. ICCS 2007, volume 4488 of Lecture Notes in Computer Science, pages 657–664. Springer Berlin /Heidelberg.

[36] Santo Fortunato and Marc Barthélemy. (2007) Resolution limit in community detection. Proceedings of the National Academy of Sciences, 104(1):36–41.

[37] Zhenping Li, Shihua Zhang, Rui-Sheng Wang, Xiang-Sun Zhang, and Luonan Chen. (Mar 2008) Quantitative function for community detection. Phys. Rev. E, 77(3):036109.

[38] Maoguo Gong, Bao Fu, Licheng Jiao. (2011) Memetic algorithm for community detection in networks. Physical Review E, 84, 056101.

[39] Dawkins, R. (1989) The selfish gene. New York: Oxford University Press.

[40] Pablo Moscato and Carlos Cotta. (2003) A gentle introduction to memetic algorithms. In Fred Glover and Gary Kochenberger,editors, Handbook of Metaheuristics, volume 57 of International Series in Operations Research & Management Science, pages 105–144. Springer New York.

[41] Krasnogor, N. and Smith, J. (Oct. 2005) A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation, 9(5):474–488.

[42] Yels, A. and Wattenberg, M. (1994) Stochastic hill climbing as a baseline method for evaluating genetic algorithms. Technical report, No. UCB/CSD-94-834. University of California, Berkeley.

[43] Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science 220 (4598): 671–680. doi:10.1126/science.220.4598.671. JSTOR 1690046. PMID 17813860.

[44] Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications 45: 41–51. doi:10.1007/BF00940812.

[45] Metropolis, Nicholas; Rosenbluth, Arianna W.; Rosenbluth, Marshall N.; Teller, Augusta H.; Teller, Edward (1953) Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics 21 (6): 1087. doi:10.1063/1.1699114.

[46] Granville, V.; Krivanek, M.; Rasson, J.-P. (1994) Simulated annealing: A proof of convergence. IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (6): 652–656. doi:10.1109/34.295910.

[47] Glover, F. and McMillan, C. (1986) The general employee scheduling problem: an integration of MS and AI. Computers and Operations Research, 13, 563-573.

[48] Fred Glover. (1989) Tabu Search - Part 1. ORSA Journal on Computing 1 (2): 190-206.

[49] Fred Glover. (1990) Tabu Search - Part 2. ORSA Journal on Computing 2 (1): 4-32.

[50] Dhillon, I. S., Guan, Y. and Kulis, B. (2004) In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM, New York), pp. 551–556.

[51] Angelini, L., Boccaletti, S., Marinazzo, D., Pellicoro, M. and Stramaglia, S. (2007) Chaos 17, 023114.

[52] Bagrow, J. and Bollt, E. (2005) A local method for detecting communities. Phys. Rev. E, vol. 72, p. 046108, 2005.

[53] White, D. R. & Harary, F. (2001) Sociol. Methodol. 31, 305–359.

[54] Albert, R., Raghavan, U. and Kumara, S. (2007) Near linear time algorithm to detect community structures in large scale networks. Phys. Rev. E 76, 036106.

[55] F. Wu and B. A. Huberman. (2004) Finding communities in linear time: a physics approach. Eur. Phys. J. B, vol. 38, p. 331.

[56] Leon Danon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas. (2005) Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment, 2005(09):P09008.

[57] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. (2008) New benchmark in community detection. arXiv:0805.4770v2 [physics.soc-ph].

[58] Zachary, W. W. (1977) An information flow model for conflict and fission in small groups. Journal of anthropological research, pages 452–473.

[59] David Lusseau, Karsten Schneider, Oliver J. Boisseau, Patti Haase, Elisabeth Slooten, and Steve M. Dawson. (2003) The bottlenose dolphin community of doubtful sound features a large proportion of longlasting associations. Behavioral Ecology and Sociobiology, 54:396–405.

[60] Newman, M. E. J. (2005) Modularity and Community Structure in Networks. In Proceedings of the National Academy of Sciences, pp. 8577–8582.