Soft Computing
https://doi.org/10.1007/s00500-018-3267-x

METHODOLOGIES AND APPLICATION

Node attraction-facilitated evolution algorithm for community detection in networks

Krista Rizman Žalik[1,2] · Borut Žalik[1]

Abstract

Network model recently has become a popular tool for studying complex systems. Detecting meaningful natural groups of nodes called communities in complex networks is an important task in network modeling and analysis. In this paper, the automatic network community detection is formulated as an optimization problem facilitated by node attraction. The basic idea is envision a network as a system of nodes where each node is attracted by its local neighbors. An evolution community detection algorithm is introduced, which employs a metric, named modularity Q as the fitness function and applies node attraction and modularity-based grouping crossover operator. The proposed algorithm faithfully captures the natural communities with high quality. Node attraction is easy to use for the speed up of the convergence of evolution algorithm to better partitions and for making the algorithm more stable. Node attraction does not require any threshold value. Experiments on synthetic and real-world networks further demonstrate the effectiveness of the proposed approach.

Keywords Community detection · Complex networks · Evolutionary algorithms · Node attraction

1 Introduction

Networks are widely used for modeling many complex real-world systems in different areas, such as sociology, biology and physics (Dorogovtsev and Mendes 2003). Networks are represented by graphs consisting of nodes and edges. Each member of a network is represented by a node, and relationship between two members of a network is represented by an edge.

In the study of complex networks, a network is said to have community structure if the nodes of the network form groups of nodes called communities, such that each group of nodes is densely connected internally. Community structure is an important property of complex systems. The study of community detection in complex networks has become a subject of research during recent years. Researchers began to unravel the structure and dynamics of complex networks (Strogatz 2001) and to study the structure and function of complex networks (Newman 2001) about 10 years ago.

In complex networks, nodes form groups of nodes called communities, such that each group of nodes is densely connected internally. Communities are groups of nodes that are densely interconnected but only sparsely connected with the rest of the network (Radicchi et al. 2004). Since for real-world networks we do not know the real number of communities, automatic clustering is required that identifies the real number of communities in networks.

During the past decades, community detection (also called graph clustering or graph partitioning) has attracted a lot of attention. Different methods have been proposed for community detection (for a review see Fortunato 2010). Many approaches have been proposed to identify communities based on different criteria (e.g., betweenness Girvan and Newman 2002), normalized cut (Shi and Malik 1997, etc.), and many of them are function optimization methods. In order to extract dense groups of nodes, an objective function can be chosen that captures the definition of a community as a group of nodes with better internal connectivity than external connectivity. Real-world networks are usually very large, and community detection in complex networks is known to be a complete NP-problem (Garey and Johnson 1979). The objective is usually NP-hard to optimize, so one usually employs

Communicated by V. Loia.

✉ Krista Rizman Žalik
krista.zalik@um.si

[1] Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia

[2] Faculty of Natural Science and Mathematics, University of Maribor, Maribor, Slovenia

Published online: 06 June 2018

heuristics (e.g., betweenness Girvan and Newman 2002) or approximation algorithms (e.g., spectral method Pothen et al. 1990).

Among function optimization algorithms, genetic algorithms (GAs) are very important approximation algorithms (Shi and Malik 1997; Pizzuti 2008) to find sets of nodes that approximately optimize the objective function and can be understood as communities. The conventional community detection can be considered as a single-objective optimization problem. A genetic algorithm (GA) is a metaheuristic using the process of natural selection, mutation and crossover to generate high-quality solutions to optimization and search problems that belongs to evolutionary algorithms (EAs). Evolutionary computation has been successfully applied to many real-world problems as an optimization technique and showed to be competitive also for the study of complex networks (for up-to-date review on evolutionary methods for community detection see Pizzuti (2017). Some evolutionary and memetic methods have been proposed to identify communities in signed networks consisting of both positive and negative links (Li et al. 2014) as well as weighted networks (Lu et al. 2015).

In this paper, the basic idea is to envision a network as a system of nodes where each node is attracted by its local neighbors. We can imagine that there is a force between each of the two connected nodes which moves them together. In this paper, the automatic community detection is formulated as an optimization problem facilitated by node attraction. Based on this formulation node attraction-facilitated evolution algorithm maximizing modularity is proposed. The proposed evolution algorithm uses a node attraction for local learning. A node belongs to community with nodes, which attracts the node more than any other neighbor community. Node attraction is easy to use for the speed up of the convergence of evolution algorithm to better partition and for making the algorithm more stable. Node attraction does not require any threshold value.

This paper is arranged as follows. Section 2 formulates the community detection problem and describes node attraction. Section 3 describes the A-Net algorithm in detail. The results of experiments of A-Net on some artificial and real-world data sets are described in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Problem definition and node attraction

Complex networks can be modeled by graphs. Let be $G(V, E)$ an undirected and unweighted graph with a set V of n nodes ($|V| = n$) and a set E of m edges ($|E| = m$). Graph adjacency matrix A describes connections of pairs of nodes by edges. Each element $A_{ij} = 1$ if there is an edge between nodes i and j and $A_{ij} = 0$ if there is no edge between nodes i and j. Community C_i is defined as a set of nodes ($C_i \in G$) with higher internal density than external density. Internal density is defined by number of links between nodes of community and external density is defined by number of edges between nodes of community and nodes belonging to other communities. Since many network partitions can satisfy this definition, we need a quality measure to measure the quality of partitions. A popular measure widely used is called modularity Q (Eq. 1) proposed by Girvan and Newman (2002). The community detection problem is to find partition with maximal modularity among all possible partitions of a given graph. This is thus a highly combinatorial optimization problem and known to be NP-hard (Brandes et al. 2008).

$$Q = \sum_{C \in X} \frac{l_c}{m} - \sum_{C \in X} \left(\frac{k_c}{2 * m} \right)^2 \tag{1}$$

where X is a partition, C is community in partition X, m is the number of edges in the whole network, l_c is the number of edges in the community C ($l_c = \sum_{i,j \in C}(A_{ij})/2$), k_C is the total degree of nodes in the C ($k_C = \sum_{i \in C}$) and k_i is the degree number of neighbors of node i. Larger value of modularity means better partitions.

2.1 Node attraction

Each node interacts with its local neighbors rather than all other nodes spreading in the network. The investigation of local neighbors of nodes can improve the modularity of partitions and quality of identified communities. Regarding any edge between two adjacent nodes, there are two interaction scenarios influencing its attraction and community assignment relaying on its topological structure: influence from directly linked nodes. A node (i) is attracted by its neighbor's node (j) with the value F_i^j. A node (j) attracts more its neighbor's node (i) when it has less neighbors: $F_i^j = \frac{A_{ij}}{k_j}$, where k_j is number of neighbors of node j . Attraction of node (i) by community (C) is denoted as F_i^C and estimates how all nodes of community (C) attract node (i).

$$F_i^C = \sum_{j \in C} \frac{A_{ij}}{k_j} \tag{2}$$

Figure 1a shows the node u that has the same number of links to two communities. It has one link to the green and one link to the red community. The green community attracts the node u more than the red community ($F_u^{\text{green}} = 1$ and $F_u^{\text{red}} = 0.34$) and the node u has to be assigned to green community. The node v in Fig. 1b has two con-

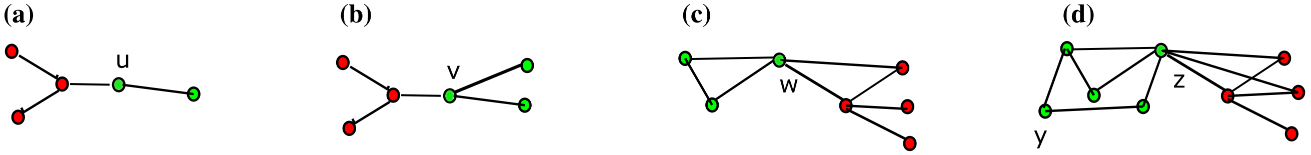


Fig. 1 Illustration of node attraction and assignment of nodes u,v,w,z to green communities in four examples (**a**)–(**d**)

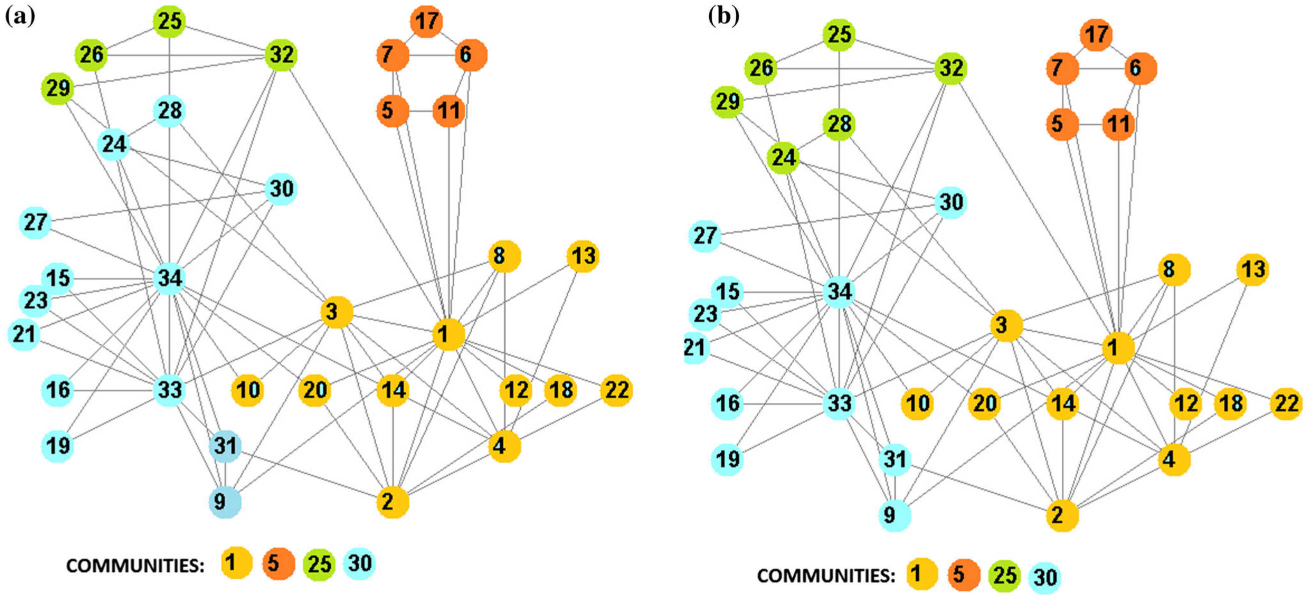


Fig. 2 Illustration of node attraction influencing the forming of green community labeled 25 on karate network described in Sect. 4.1.1: **a** before node attraction processing, **b** after node attraction processing

nections with the green community and one link with the red. It is assigned to green community ($F_v^{\text{green}} = 2$ and $F_v^{\text{red}} = 0.34$). A node w in Fig. 1c and a node z in Fig. 1d has the same number of links to both communities (the red and the green), but both nodes are assigned to the green community, because attraction value of green nodes is higher.

Here we consider the second scenario: the influence from the common neighbors. Nodes z and y in Fig. 1d have three common neighbors. As the common neighbors have both links with the two nodes z and y, they attract the two nodes and will result in increase of attraction value.

Nodes move among community until all nodes belong to community with the greatest attraction. First the node 24 is checked in the example in Fig. 2a and the node 24 stay in community 34 with the greatest attraction ($F_{24}^{25} = 0,333333333$; $F_{24}^{34} = 0,558823529$). Community structure after node attraction process is shown in Fig. 2b. Next node 28 is moved to community 25 ($A_{28}^{25} = 0,333333333$; $A_{28}^{34} = 0,258823529$) and then node 24 is moved to community 25 ($A_{24}^{25} = 0,583333333$; $A_{24}^{34} = 0,308823529$).

3 Description of A-Net

We choose the modularity as the optimization function of our genetic algorithm (GA) for community detection. GAs work with more solutions that form population and each solution is an individual or a chromosome. At each iteration of a GA, a new population is generated using several operators: reproduction, crossover and mutation.

First, initial population is generated. Then we use crossover that maximally increases modularity described in Sect. 3.2. We use a neighbor-based mutation operation which takes into account the effective connections and enters new genetic material to population. Then the local learning phase improves the solution population by merging communities to increase modularity and reassign nodes to the community with the greatest attraction. The group-based representation of chromosomes described in the next section is used.

3.1 Representation

Group-based representation (Tasgin et al. 2007) is commonly used representation of community partitions in genetic algorithms. Each solution represents population of N chromosomes X_p; $1 \leq p \leq N$. Each chromosome X_p has a set of values (genes) x_p^i; $1 \leq i \leq n$, where n is the number of all nodes. Each chromosome $X_p = \{x_p^1, x_p^2, \ldots, x_p^n\}$ has n genes. ith gene x_p^i contains value from range of 1 to n which denotes the community identification number (community label) of node i in partition X_p. Values are initialized at the beginning and are changed by crossover or mutation operators or local learning. All nodes with the same genes' community identification number form one community. The number of different community labels in resulted partition automatically determines the number of communities.

3.2 Crossover operators

Crossover is an important method of optimization in the GA that mainly defines the performance of the GA. We use crossover that maximally increases modularity. The input to crossover operator is two chromosomes (parents), and the result is one or two chromosomes (child). The crossover operator should inherit their good parent chromosomes' communities and add new ones. Crossovers that provide intelligent evolution require additional computation cost compared to classic crossover operators.

3.2.1 Crossover based on ratio of modularity and the number of nodes in community

We use crossover that maximally increases modularity. In Rizman Žalik and Žalik (2017), we evaluated more crossover operators, and the highest increase of modularity is obtained when crossover based on ratio of modularity and the number of nodes in community is used. We sort the communities of both chromosomes (parents) by the ratio of modularity and number of nodes assigned to community. The community with the highest ratio is formed first in the offspring. Then the community with the second highest ratio of modularity is tried to be formed in the offspring, but only from nodes that are not assigned to any community that was formed before. Forming of communities is stopped when all genes of offspring are assigned to communities.

Many communities with only a few or even a single node can be generated. These small communities are not real communities and should be allocated to the other communities.

3.3 Detailed steps for the A-Net

Algorithm 1 shows the pseudo code of the DPA. The non-dominated sorting genetic algorithm-II (NSGA-II) (Srinivas and Deb 1994) is adopted as optimization mechanism. Basically, A-Net begins with an initial population of solutions and then repeats an iterative process of crossover, mutation, local learning and selection for a number of times (generations). Mutation randomly changes genes of one chromosome and includes new genetic material in offspring population. The crossover operator is applied to two parents that are randomly chosen from offspring population. Finally, offspring solution is inserted into the population and the best chromosomes are selected and included in new population. The algorithm stops when the number of generations reach the given threshold. The detailed process is described as follows. *Step 1* Initialization. Generate initial population P_0 of N chromosomes. ith gene corresponds to the ith node of graph with values $1 \ldots n$, where n is the number of nodes in graph. The value of gene is a community label.

Algorithm 1: Algorithm A-Net

```
Data: Graph $G = (V, E)$, maximum number of generations
      $g_{\max}$, population size $N$, mutation probability $p_m$,
      crossover probability $p_c$
Result: Set of partitions $P = \{X_1, X_2, \ldots, X_p, X_{maxQ}\}$ with
        partition $X^*$ with a maximal modularity.
1. Initialization:
$P_0 = \{X_1, X_2, \ldots, X_N\} \leftarrow$ Initialize Population(G,N)
2. Calculate values of modularity for all chromosomes. Repeat in
more iterations; $k$ counts the number of iterations.
$k \leftarrow 0$
repeat
    $k \leftarrow k + 1$;
    3. Mutation and crossover are performed according to the
    predefined ratio (mutation probability $p_m$, crossover
    probability $p_c$) on the randomly chosen solutions. Offspring
    population $P^o$ is generated:
    $P^o \leftarrow$ clone($P_{k-1}$)
    if rand(0..1) $> p_c$ then $P^o \leftarrow$ Q/n-crossover operator($P^o$)
    if rand(0..1) $> p_m$ then $P^o \leftarrow$ mutation operator($P^o$)
    4. Local learning:
    Reasign the nodes to the communities with the highest values
    of community attraction values.
    Merge communities to increase the modularity.
    5. Form the population $P_k$:
    $P_k \leftarrow$ keep the $N$ best chromosomes in the population
    $P_{k-1} + P^o$.
    Identify $X_{\text{maxQ}}$; return $X_{\text{maxQ}}$
until $k < g_{\max}$;
```

Step 2 Calculate the modularity function values of the chromosomes and rank them.

Step 3 Apply proportional cloning to population P_{t-1} and perform the usual binary tournament selection, mutation and modularity crossover operator on the clone population to create an offspring population P^o.

Step 4 Repair the child chromosomes. As stated in Sect. 3.2.1, using modularity-based crossover causes generation of many communities with only a few or even a single node. These small communities are not real communities and should be allocated to other communities, which are accomplished by repairing the child chromosome.

Step 4.1 Merge communities to increase modularity. Repeat the process until the modularity cannot be more increased.

Step 4.2 Reassign nodes to the community with the greatest attraction.

Then we improve the quality of communities by satisfying attraction constraint discussed in Sect. 3.2.1. First we rank the neighbor communities by their attraction values. All nodes are moved to the community with the highest attraction. Repeat the process of node moving for all chromosomes.

Step 5 Model selection. First, a combined population is formed $R_t = P_t + Q_t$. Each solution is assigned a fitness (or rank) equal to its nondomination level (1 is the best level, 2 is the next-best level, and so on). Thus, minimization of fitness is assumed. Then, the population R_t is sorted according to nondomination. Crowding distance values of all nondominated chromosomes is calculated, and first N chromosomes are chosen as new population. Dominant population is updated.

Repeat all steps from Step 2 until the iteration number reaches $g_{\max}$.

3.4 Initialization

Population initialization is the first task in evolutionary algorithms. It is important task, since the initial population can affect the convergence speed and also the quality of the final solution. Random initialization is used to generate initial population, while usually no information about solution exists. Initially, each node is put in a different community by initialization of each gene (x_p^i; $i = 1, 2, \ldots, n$) of each (pth) chromosome (X_p; $p = 1, 2 \ldots, N$) in the initial population P_0 with value i: *i.e.*, $x_p^i = i$. Then for each node i randomly select one neighbor node k in the network and set value of gene $x_p^i = k$. For each node of population, the neighbor is chosen completely randomly, which generates unsupervised solutions, although some researches have shown the relevant community structure can be revealed exploiting maximal neighbor similarity (Rizman Žalik 2015). So locus-based adjacent representation of initial population is formed, where each gene shows to one node's neighbor. Then group-based representation of initial population is obtained. Each gene x_p^i is assigned the value of a gene x_p^k, where k is randomly chosen neighbor of node i in the network. The proposed initialization process takes into account the effective connections of the network, and the generated solutions are unsupervised.

3.5 Crossover based on the ratio of modularity and number of nodes in community

We use crossover to inherit the best communities with the highest ratio of modularity and number of nodes in community from one generation to the next. First, tournament selection is used for selecting parents for crossover operation. Then we sort all communities from both parents by the ratio of modularity and number of nodes assigned to community. Then we start to generate child chromosomes in the offspring. First we create the community that is the same to the community with the highest ratio. Then we create the community with the second highest ratio in the offspring, but only from nodes not assigned before. We stop when all nodes in the offspring are assigned. A lot of small communities with even one or two elements can appear after crossover operation, which are not real. Merging of connected communities can increase modularity. A multistep greedy agglomerative algorithm (MSG) (Schuetz and Caflish 2008) is used. It can merge multiple pairs of communities at a time. Community pairs (i, j) are joined when the modularity change is the greatest and positive and neither community i nor community j is presented in another pair inducing the highest modularity change.

3.6 Mutation

Mutation generates random changes in chromosomes and inserts a new genetic material to population. Mutation is used to prevent the results to stuck in local extreme. The offspring is created by randomly changing part of existing chromosomes. Mutation is not performed often, because it causes random search and large-scale mutations usually produce chromosomes that do not survive the selection. We use a neighbor-based mutation operation which takes into account the existing connections (Pizzuti 2012). We perform the mutation on each node of chromosome if a generated random value $r \in [0, 1]$ is smaller than the mutation probability p_m. Then the community label of this node is replaced with one of its neighbors' label by assigning gene $x_a^i \leftarrow x_a^j, \ \exists j \in \{j | A_{ij} = 1\}$.

3.7 Local learning

In each generation, local search procedure is applied to chromosomes in population to obtain better chromosomes. We use here attraction of neighbor nodes. Attraction values (see Eq. 2) to all neighbor communities are calculated for each node on the border and the highest attraction value defines the community to which the node is moved.

4 Experiment results

In this section, extensive experiments are done on some artificial and some real-world networks that are often used for evaluation of new methods. Modularity is used as a quality

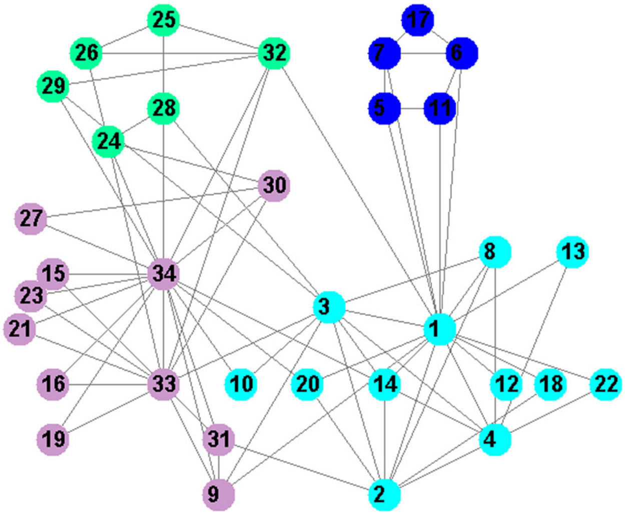


Fig. 3 Zachary karate club network

measure for evaluation of obtained partitions of real-world networks.

4.1 Simulation on some typical networks

4.1.1 Zachary karate club network

Zachary's Karate Club is a well-known social network. The network captures 34 members of a karate club and 78 pairwise links between members who interacted outside the club (Zachary 1977). Four communities are identified in the resulted partition with modularity 0,4198 shown in Fig. 3.

4.1.2 Dolphin network

An undirected social network of frequent associations between 62 dolphins in a community living in New Zealand (Lusseau et al. 2003). Five communities are identified in the resulted partition with modularity 0,5285 shown in Fig. 4.

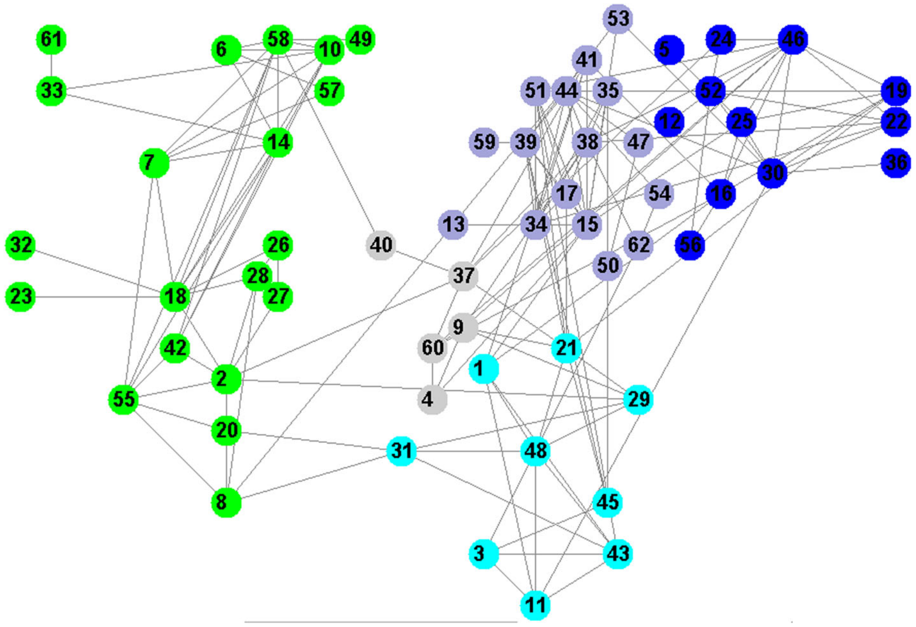
4.1.3 College football network

Network of American football games between Division IA colleges during regular season Fall 2000 (Girvan and Newman 2002). Ten communities are identified in the resulted partition with modularity 0,6058 shown in Fig. 5.

4.2 Comparison with other algorithms

We compared results with three other community detection algorithms. Meme-Net Gong et al. (2011) has been proposed to optimize another quality function, modularity density. Meme-Net uses a tunable parameter that allows identifying structure of the network at different resolutions. The algorithm combines a genetic algorithm with a hill-climbing strategy to optimize the general version of modularity density. It was proposed to detect communities at different hierarchical levels. The Multi-objective Genetic Algorithm for Networks (MOGA-Net) (Pizzuti 2012) optimizes community score and community fitness simultaneously. It adopts the nondominated sorting genetic algorithm-II (NSGA-II) Deb et al. (2002) as optimization mechanism. MMCD (Wu and Pan 2015) is memetic multi-objective optimization method which uses the two contradictory parts of modularity function as two objective functions. For local search procedure, a new

Fig. 4 Dolphin network



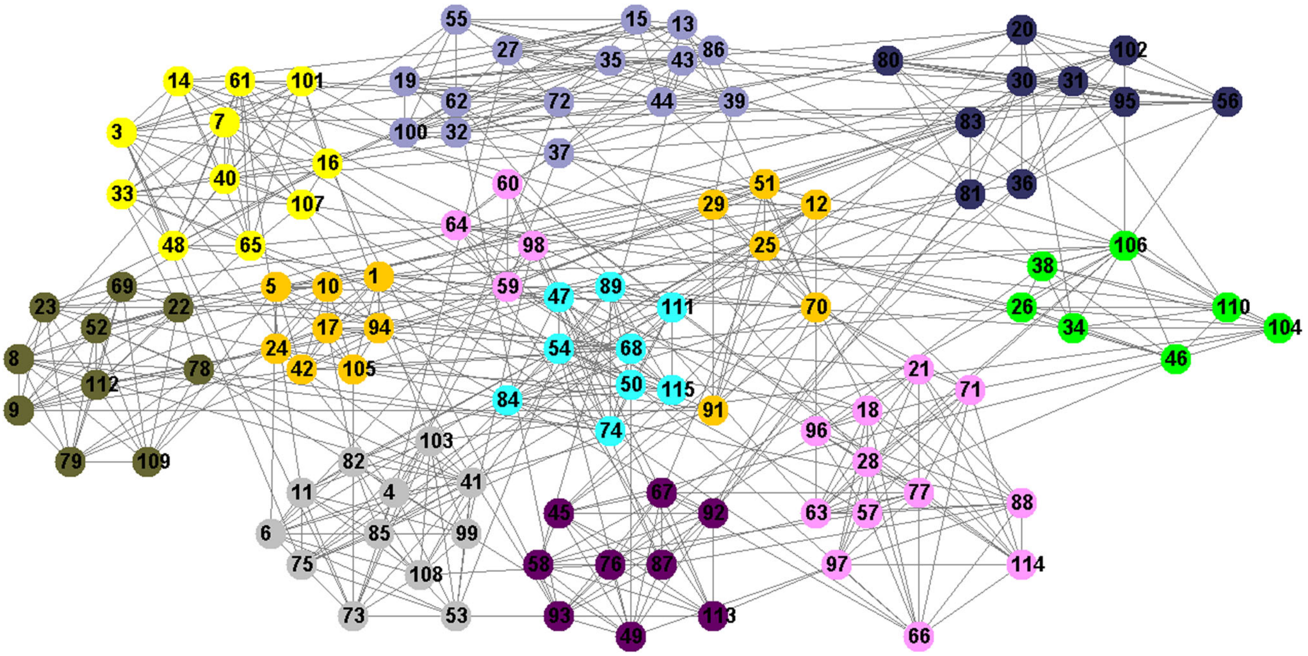


Fig. 5 Football network

Table 1 The maximal modularity ($Q_{\max}$), average modularity (Q_{avg}) and standard deviation of modularity ($Q_{\max}$) obtained by A-Net compared to obtained values of MMCD, MOGA-Net and MEME-Net and for 5 real-world data sets: karate, dolphins, polbooks, football and jazz

Network	Criterion	A-Net without local search	MMCD	MOGA-Net	MEME-Net	A-Net
Karate	$Q_{\max}$	0.4198	0.4159	**0.4198**	**0.4198**	**0.4198**
	Q_{avg}	0.4175	0.3946	0.4194	0.4191	**0.4198**
	Q_{std}	0.0026	0.0083	0.0019	0.0015	**0**
Dolphins	$Q_{\max}$	0.5277	0.5034	0.5185	**0.5285**	**0.5285**
	Q_{avg}	0.5190	0.4589	0.5096	0.5269	**0.5285**
	Q_{std}	0.067	0.0093	0.0061	0.0014	**0**
Polbooks	$Q_{\max}$	0.5244	0.4999	0.5232	**0.5272**	0.5267
	Q_{avg}	0.5037	0.4625	0.5218	**0.5271**	0.5247
	Q_{std}	0.0165	0.0120	0.0031	0.0008	**0.0001**
Football	$Q_{\max}$	0.5894	0.4329	0.6044	0.6046	**0.6058**
	Q_{avg}	0.5683	0.3960	0.6022	0.6040	**0.6057**
	Q_{std}	0.0158	0.0169	0.0016	0.0006	**0.00005**
Jazz	$Q_{\max}$	0.4143	0.2929	0.4421	**0.4451**	0.4421
	Q_{avg}	0.3980	0.2925	0.4419	**0.4449**	0.4416
	Q_{std}	0.0094	0.0080	**0.0001**	0.0002	0.0003

The best values in each row are bold

direction vector is used to integrate two objective functions together and then the local optimal solutions are found by a network specific local search strategy based on label propagation rule.

Modularity values for some real-world networks, that are often used as a test bad for new community detection methods, are compared to modularity values obtained by A-Net in Table 1.

The results shown in Table 1 for considered real-world networks prove the efficiency of the proposed method as the modularity values obtained by A-Net are the largest in the compared algorithms. Bold number in each row denotes the best value. A-Net obtained the best values in 10 items of all 15 items. We run the A-Net algorithms for 10 times independently on each network. The parameters of algorithm A-Net are following: population size 100, maximum generation number $g_{\max} = 5$ crossover probability p_c=0.8, mutation probability $p_m = 0.2$.

The results shown in Table 1 also confirm that A-Net performs better, because of performing local search using nodes' attractions. Modularity values using only mutation and crossover based on ratio of modularity and number of communities without local search are much lower than obtained by A-Net (Table 1).

4.2.1 Comparison on artificial networks

Since the community structure of generated artificial networks is known, it is easy to get the similarity between the solution obtained by an algorithm and the ground truth. For this purpose, we use common quality measurement normalized mutual information (I) (Danon et al. 2005). I is defined as follows. A and B are partitions of a network, where A corresponds to the real communities and B corresponds to the predicted communities. N_A represents the number of communities in A and N_B denotes that of B, S is matrix with elements S_{ij} counting the number of nodes in community i of A that also appear in community j of B. N is the number of elements. $Si.$ is the sum over row i of S while $S.j$ is the sum of elements in column j. The definition of $I(A, B)$ is shown as:

$$I(A,B) = \frac{-2\sum_{i=1}^{N_A}\sum_{j=1}^{N_B} S_{ij}\log\left(\frac{S_{ij}N}{S_{i.}S_{.j}}\right)}{\sum_{i=1}^{N_A} S_{i.}\log\left(\frac{S_{i.}}{N}\right)+\sum_{j=1}^{N_B} S_{.j}\log\left(\frac{S_{.j}}{N}\right)} \quad (3)$$

We have evaluated the accuracy of the proposed algorithm on some artificial networks GN (Girvan and Newman 2002). The GN networks consist of 128 nodes which are divided into four communities with 32 nodes each. Each node has an average degree of 16 and shares $(1-\mu)$ edges with the nodes of its community. μ is called the mixing parameter. When μ>0.5, the average number of neighbors of each node in its community is smaller than that of neighbors belonging to the rest communities and the network has weak community structure. The higher the mixing parameter of a network, the less separated are communities and more difficult for identifying. The comparisons of average values of I obtained by A-Net with reported for two other compared methods are shown in Fig. 6. A-Net has a good performance in the artificial GN networks.

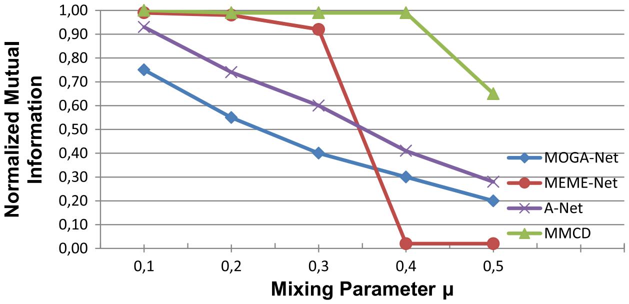


Fig. 6 Performance comparisons between A-Net and published MMCD, Meme-Net and MOGA-Net on the GN benchmark networks with different mixing parameters μ ranging from 0.1 to 0.5

5 Conclusion

In this paper, the automatic network community detection is formulated as an optimization problem facilitated by node attraction. Based on this formulation, the evolution algorithm maximizing the modularity criterion function is proposed. Furthermore, with using node attraction for local learning, the accuracy of the community detection increases. Node attraction accelerates convergence to an optimum. The efficiency of the proposed algorithm is verified by comparative experiments on real world and synthetic GN benchmark networks.

There are many possible directions for future work. A-Net can be extended with incorporating other sources of information that influence node attraction like node attributes or edge attributes.

Acknowledgements This work was supported by the Slovenian Research Agency (grant numbers: P2-0041, J2-8176).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Brandes U, Delling D, Gaertler M, Gorke R, Hoefer M, Nikoloski Z, Wagner D (2008) On modularity clustering. IEEE Trans Knowl Data Eng 20(2):172–188

Danon L, Diaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. J Stat Mech: Theory Exp 2005(09):P0900

Deb K (2001) Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester

Deb K, Pratap A, Agarwal SA, Meyarivan T (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Dorogovtsev SN, Mendes JFF (2003) Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press, Oxford

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3):75–114

Fortunato S, Barthélemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci U S A 104(1):36–41

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP completeness. W. H. Freeman, New York

Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99:7821–7826

Girvan M, Newman ME (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826

Gong M, Fu B, Jiao L, Du H (2011) A memetic algorithm for community detection in networks. Phys Rev E 84(5):05610

Gong M, Chen X, Ma L, Zhang Q, Jiao L (2013) Identification of multi-resolution network structures with multi-objective immune algorithm. Appl Soft Comput 13:1705–1717

Hu Y, Yang Bo, Wong H-S (2016) A weighted local view method based on observation over ground truth for community detection. Inf Sci 355–356:37–57

Li Y, Liu J, Liu C (2014) A comparative analysis of evolutionary and memetic algorithms for community detection from signed social networks. Soft Comput 18(2):329–348

Lu Z, Sun X, Wen Y, Cao G, La Porta T (2015) Algorithms and applications for community detection in weighted networks. IEEE Trans Parallel Distrib Syst 26(11):2916–2926

Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54:396–405

Ma L, Gong M, Liu J, Cai Q, Jiao L (2014) Multi-level learning based memetic algorithm for community detection. Appl Soft Comput 19:121–133

Newman MEJ (2001) The structure and function of complex networks. SIAM Rev 2(45):167–256

Park Y, Song M (1998) A genetic algorithm for clustering problems. In: Proceedings of the third annual conference on genetic programming. pp 568–557

Pizzuti C (2008) Ga-net: a genetic algorithm for community detection in social networks. In: PPSN. pp 1081–1090

Pizzuti C (2012) A multiobjective genetic algorithm to find communities in complex networks. IEEE Trans Evol Comput 16:418–430

Pizzuti C (2017) Evolutionary computation for community detection in networks: a review. Trans Evolut Comput. https://doi.org/10.1109/TEVC.2017.2737600

Pothen A, Sinmon H, Liou K-P (1990) Partitioning sparse matrices with eigenvectors of graphs. SIAM J Matrix Anal Appl 11:430–452

Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D (2004) Defining and identifying clusters in networks. Proc Natl Acad Sci USA 9(101):2658–2663

Rizman Žalik K, Žalik B (2017) Multi-objective evolutionary algorithm using problem-specific genetic operators for community detection in networks. Neural Comput Appl. https://doi.org/10.1007/s00521-017-2884-0

RizmanŽalik K (2015) Maximal neighbor similarity reveals real communities in networks. Sci Rep 5:1837

Schuetz P, Caflish A (2008) Efficient modularity optimization by multistep greedy algorithm and node refinement. Phys Rev E 77(4):046112

Shi J, Malik J (1997) Normalized cuts and image segmentation. IEEE Trans PAMI 22:888–905

Shi C, Yan Z, Cai Y, Wu B (2012) Multi-objective community detection in complex networks. Appl Soft Comput 12:850–85

Srinivas N, Deb Kalyanmoy (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. Evolitionary Computation 2(3):221–248

Strogatz SH (2001) Exploring complex networks. Nature 410:268–276

Tasgin M, Herdagdelen A, Bingol H (2007) Community detection in complex networks using genetic algorithms. arXiv: 0711.0491

Wu P, Pan L (2015) Multi-objective community detection based on memetic algorithm. PloS ONE 10:e0126845

Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33:452–473

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.