Information Sciences 445–446 (2018) 38–49

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Memetic algorithm using node entropy and partition entropy for community detection in networks

Krista Rizman Žalik[a,b,*], Borut Žalik[b]

[a] Faculty of Electrical Engineering and Computer Science, Faculty of Natural Science and Mathematics, University of Maribor, Slovenia
[b] Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia

ARTICLE INFO

Article history:
Received 21 June 2016
Revised 25 February 2018
Accepted 26 February 2018
Available online 27 February 2018

Keywords:
Community detection
Complex networks
Evolutionary algorithms
Memetic algorithms
Entropy
Modularity

ABSTRACT

Community detection is a key to understanding the structure of complex networks. Many community detection approaches have been proposed based on the modularity optimization. Algorithms that optimize one initial solution often get into local optima, but algorithms that simultaneously optimize a population of solutions have high computational complexity. To solve these problems, genetic algorithms improved by a local learning procedure known as memetic algorithms can be applied. We propose a memetic algorithm for community detection in networks, that exploits node entropy for local learning. Node entropy is easy to use to speed up the convergence of an evolutionary algorithm and to increase the quality of partitions, while it uses only the node's neighborhood and does not require any threshold value. Moreover, this algorithm is slightly modified in order to avoid modularity function which suffers a resolution limit and, therefore, it may fail to detect small communities. We propose and use an entropy function as an optimization function and as criteria in grouping crossover operator. Experiments on real-world and synthetic networks illustrate that the proposed method can find natural partitions effectively.

1. Introduction

Complex real-world systems in biology, physics, sociology and other areas, which are composed of many components and interactions among them, are widely modeled with networks [6]. Most real-world networks, such as computer networks, information networks, friendship networks, technological networks, protein networks or biological networks, are complex. Networks are represented by graphs consisting of edges, also called links, and nodes, called vertices. Each member of a network is represented by a node and relationship between two members of a network is represented by an edge between two nodes. Some nodes are connected more densely and form groups, called also modules or communities. Community structure is an important property of complex systems as uncovering of partitions of complex networks helps to understand their structure and functionality [34]. Since we do not know the real number of communities, automatic clustering is required for uncovering natural communities and number of existing communities in complex networks. No common definition of communities is agreed. Different definitions are used in different community detection methods. The often used definition requires that there are more edges inside a community than there are outside edges that connect nodes of the commu-

* Corresponding author.
E-mail address: krista.zalik@um.si (K.R. Žalik).

https://doi.org/10.1016/j.ins.2018.02.063
0020-0255/

nity to other communities. It defines communities as groups of nodes that are interconnected densely but only connected sparsely with the rest of the network [27].

During the past decades, different methods have been proposed for community detection also called graph clustering or graph partitioning (for a review see [8]). Different criteria (e.g. normalized cut [33], betweenness [22] etc.) have been proposed to identify communities. Many community detection approaches have been proposed based on the modularity optimization [11], which suffers the resolution limit [9], while it is unable to detect small communities even when they are well-separated. Modularity has a resolution limit of $\sqrt{|E|}$ order. Communities with the number of edges inside community smaller than $\sqrt{|E|}/2$ cannot be uncovered using the modularity optimization. The multiple resolution approach has been proposed and studied [19]. For small resolution parameters, it tends to group together small communities. And when the resolution is high, it splits large communities. Thus, either smaller or larger communities are affected by the resolution limit for each chosen resolution limit. Resolution limit problem can be solved by slightly different quality measures such as modularity density [2]. Newman's modularity has also been used for identifying overlapping (fuzzy) communities [17]. Real world networks are usually very large, and community detection in complex networks is known to be a complete NP-problem [10]. Therefore, approximation algorithms, such as spectral method [26]) or evolutionary algorithms (EAs), are used for community detection [24,33]. Evolutionary algorithms exploit the principles of natural evolution, such as selection of the best individuals from population, recombination, and mutation. Fitness function is used to evaluate the individuals of population required for selection. Evolution algorithms which can optimize more optimization functions are multi-objective evolutionary algorithms [4]. MOCD [33] is a multi-objective evolutionary algorithm which optimizes two terms of modularity. The Multi-Objective Genetic Algorithm for Networks (MOGA-Net) [25] optimizes community score and community fitness simultaneously. It uses the Nondominated Sorting Genetic Algorithm-II (NSGA-II) [5] as an optimization mechanism. Meme-Net [13] uses modularity density quality function for uncovering communities at different hierarchical levels. Meme-Net uses a tunable parameter for uncovering the structure of the network at different resolutions and combines a genetic algorithm with a hill-climbing strategy. Genetic algorithms are efficient as a global search technique, but they need a relatively long time to converge to a global optimum. Genetic algorithms that simultaneously optimize a population of solutions have high computational complexity, but algorithms which optimize one initial solution often get into local optima. Memetic algorithms with integrating local learning into genetic procedure, solve both problems. Population-based hybrid Genetic Algorithms (GAs) extended with an individual learning mechanism for local improvements of solution are called Memetic Algorithms (MAs) [21]. Genetic algorithm is more suited to find good solutions quickly than to find the absolute best solution while local learning is efficient in finding the absolute optimum in narrow solution space. The local search improves some best individuals of the population and so speeds up the overall optimization process.

In community detection, moving nodes to the most adequate community, improves the fitness of solution and accelerates the convergence of genetic algorithm. For searching the most adequate community for each node, only local node surrounding (direct neighbors) can be observed. Local learning in the community detection process integrates some knowledge about community detection into genetic algorithm which makes the memetic algorithm more efficient for the community detection problem than genetic algorithms. The local learning mechanism is the most important for quality of the memetic algorithm because it defines the ability of local convergence. Because entropy measures the impurity, we use it in local learning. Node entropy is easy to use to speed up the convergence of an evolutionary algorithm and to increase the quality of partitions, while it uses only the node's neighborhood and does not require any threshold value. In this paper we propose an algorithm Node Entropy MA for Networks (NE-Net) using the node-entropy local learning procedure. The proposed algorithm uses modularity as a fitness function and problem specific genetic operators: modularity based group crossover and mutation. To show the superiority of our proposed algorithm over the some well-known algorithms, several real-world data sets have been used in evaluation. Experiments on computer-generated and real-world networks show the effectiveness of our algorithm.

Moreover, this algorithm is slightly modified in order to avoid modularity function which suffers a resolution limit and to make the proposed algorithm able also to identify small communities in cases where communities are unambiguously defined. We propose and use an entropy function as an optimization function and as criteria in groping crossover operator. We name this slightly modified algorithm Entropy based MA for Networks (E-Net).

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

2. Problem definition

Let us denote $G(V, E)$ an unweighted, undirected graph, where V ($|V| = n$) represents the set of nodes also called vertices and $E = V \times V$ ($|E| = m$) the set of edges. Let us denote with A the adjacency matrix of the network G, where $A_{ij} = 1$ if there is an edge between nodes i and j and 0 otherwise. A community (C_i) consists of nodes clustered into tightly connected groups forming subgraphs ($C_i \in G$) with a high density ($E_{C_i}^{in} = \sum_{i,j \in C_i} A_{ij}$) and a lower density of between the group connections ($E_{C_i}^{out} = \sum_{i \in C_i, j \notin C_i} A_{ij}$). This definition of community can be satisfied by more partitions having different or the same numbers of communities. Therefore a quality measure is necessary for detection of the best partition among all the

possible partitions of a given graph. Community detection methods have to maximize the quality metric over all possible partitions of a network.

The widespread quality measure is modularity Q which is based on the idea that nodes of community have more internal links than randomly connected nodes. It compares the fraction of edges within the community with the fraction of randomly connected nodes [11].

$$Q = \sum_{C_i \in P} \frac{E_{C_i}^{in}}{m} - \left(\frac{2 * E_{C_i}^{in} + E_{C_i}^{out}}{2 * m} \right)^2 \tag{1}$$

where m is the number of all edges, $E_{C_i}^{in}$ is the number of edges between nodes of a community C_i and $E_{C_i}^{out}$ is the number of edges between nodes of community C_i and nodes from other communities.

The other form of modularity is

$$Q = \sum_{C_i \in P} \frac{A_{ij}}{2 * m} - \left(\frac{k_i * k_j}{4 * m} \right)^2 \delta_{c_i, c_j} \tag{2}$$

where δ denotes the Kronecker Delta, which is 1 if its arguments are identical, and 0 otherwise, c_i and c_j denotes community labels of nodes i and j respectively, k_i is degree—the number of neighbors of node i and k_j is degree of node j, m is the number of edges in the whole network. The modularity is often used for clustering [1]. We use modularity as the optimization criteria, because it is the most often used optimization function in community detection.

In this paper we also propose to use node entropy for local learning described in the next subsection.

2.1. The node entropy

In information theory, Shannon entropy [32] is the expected average value of the information contained in each message or information. We use a node entropy derived from Shannon entropy as a measure of the node's similarity to the current community. The original meaning of Shannon information is a measure of uncertainty information of a probability distribution, but it is also used to denote disorder or discrepancy and equality [18].

Each network node can be allocated to more communities, because its neighbors can belong to more communities. A node can join to any community of one of its neighbors. Node entropy is measurement of uncertainty about the node and natural community.

The node entropy of node i is the scaled Shannon entropy of the weights of all edges adjacent to node i by logarithm of degree of node i:

$$NE(i) = -\frac{\sum_{j \in \Gamma_i} p_i^j \log(p_i^j)}{\log(deg(i))}; \; p_i^j = \frac{1}{deg(j)}; \tag{3}$$

where degree of node i is the number of edges adjacent with node i in the neighborhood and Γ_i neighborhood of node i:

$$\Gamma_i = \{ j \in V | \{i, j\} \in E \} \tag{4}$$

The part of node entropy caused by nodes from the community C is

$$NE_C = -\frac{\sum_{i \in C, j \in \Gamma_i} p_i^j \log\left(p_i^j\right)}{\log(deg(i))}; \; p_i^j = \frac{1}{deg(j)}; \tag{5}$$

To determine the community to which each node belongs the following calculations are performed. The entropy is calculated for each community (C) to which a node could possibly join. The node can join to any community to which belongs one or more nodes' direct neighbors. The entropy is also calculated for the community that the node currently belongs to. Each node is assigned to the community C with the greatest part of node entropy NE_C.

3. Description of NE-Net

GAs and MAs work simultaneously on more solutions forming populations of chromosomes, which represent possible solutions. Every chromosome consists of n genes, each of which represents one node of graph by containing label of community to which a node is assigned or value of index of one neighbor. Assuming a serial numbering of nodes starting with 1, gene i corresponds to the node n_i. Commonly used representation of chromosomes in evolutionary algorithms is group based representation [35]. Each gene of chromosome contains a value of the group to which it belongs. In locus based adjacent representation [23] each gene contains an index of one of node's neighbors. In adjacent locus based representation all nodes with genes and their adjacent genes form a community, while in group based representation all nodes represented by genes with the same value of community label form a community. We use group based representation. All nodes with the same community label form one community automatically. The number of different community labels in the resulted partition automatically uncovers the number of communities in the network. GAs build a solution in more iterations using several genetic operators: selection, reproduction, crossover and mutation. We use a problem-specific crossover operator.

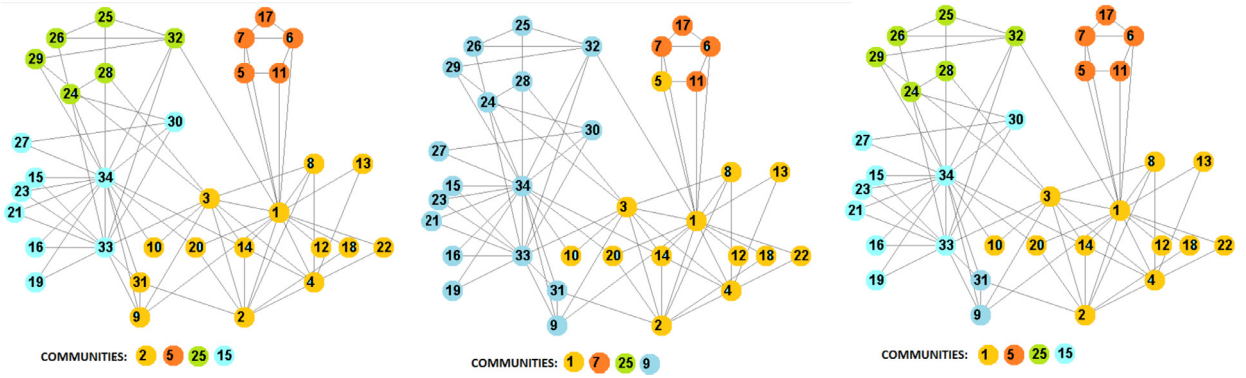

Fig. 1. Illustration of crossover on the karate network described in Section 4.1.1: a) Parent 1, b) Parent 2, c) Offspring.

3.1. Crossover operators

Crossover operator mainly defines the performance of the GA. Two individuals (parents) are input to each crossover operator and one individual (child) or two are output. Crossover inherits good communities from parent chromosomes and adds new ones. Many crossover operators have been proposed. Uniform crossover [24] uses a randomly generated binary vector. For value 0 of the binary vector, the solution genes are taken from the first chromosome and for value 1 of the binary vector from the second chromosome. In two-point crossover [33] the cluster identifiers between the two chosen nodes in the parent chromosomes are swapped and two child chromosomes are generated. The uniform crossover and the two-point crossover are rarely efficient while they produce very random offspring. But they are faster in comparison with crossovers that provide intelligent evolution which require additional computation.

3.1.1. Crossover based on the ratio of modularity and the number of nodes in a community

The goal is to use crossover that inherits the best communities from parents and increases modularity as much as possible. Communities with the highest ratio of modularity and the number of nodes of community increase modularity of the whole partition more than communities with the highest modularity [29]. We sort the communities of both parent chromosomes by the ratio of modularity and number of nodes assigned to the community. The same communities are tried to be formed in the offspring as the parents have, but only from nodes that were not allocated to any community formed before. First a community with the highest ratio is formed in the offspring and then with the second highest ratio and so on. When all genes of offspring are assigned to communities, the process stops. 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. Values of ratio Q/n_c for all communities of both parents shown in Fig. 1 and Table 1 ordered by ratio Q/n_c:

Q/n_c(community 5 parent 1)=0.0133,
Q/n_c(community 15 parent 1)=0.0121,
Q/n_c(community 7 parent 2)=0.0115,
Q/n_c(community 1 parent 2)=0.0111
Q/n_c(community 25 parent 1)=0.0110,
Q/n_c(community 9 parent 2)=0.0109
Q/n_c(community 2 parent 1)=0,0108.

The order of forming of communities in the offspring shown in Table 2 is: community 5 from the first parent, community 15 from the first parent, community 7 of the second parent cannot be formed (all its nodes were assigned before to community 5), community 1 of the second parent is formed and then community 25 of the first parent and community 9 of the second parent. The resulted partitions contain small communities that can be merged to increase the modularity. Only community 9 with 3 elements and community 15 can be merged to increase modularity. This merge increases the modularity of the offspring to 0.4198, while the modularity of the first parent is 0.3934 and of the second parent is 0.3764.

3.2. Detailed steps of the NE-Net community detection

Steps of the algorithm NE-Net are shown in Algorithm 1. The algorithm minimizes the used optimization function modularity. The Nondominated Sorting Genetic Algorithm-II (NSGA-II) [31] is used as an optimization mechanism.

After initialization, an iterative process of genetic operations is performed on population and selection of a new population for a number of times (generations). Crossover and mutation operators are performed at each generation. The crossover operator is applied to two parents that are chosen randomly from the offspring population. Finally, the offspring population

Table 1
Two parents with 34 nodes numbered from 1 to 34. Communities of the first parent: 2,5,15,25 and of the second 1,7,9.

Parent	Node number																																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1.	2	2	2	2	5	5	5	2	2	2	5	2	2	2	15	15	5	2	15	2	15	2	15	25	25	25	15	25	25	15	2	25	15	15
2.	1	1	1	1	1	7	7	1	9	1	1	1	1	1	9	9	7	1	9	1	9	1	9	9	9	9	9	9	9	9	9	9	9	9

Table 2
Offspring creation process using parents from Table 1, crossover based on ratio of modularity and number of nodes, and merging of all communities for increasing modularity.

Step	Node number																															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1.					5	5	5				5						5																	
2.					5	5	5				5				15	15	5		15		15		15				15						15	15
3.	1	1	1	1	5	5	5			1	5	1	1	1	15	15	5	1	15	1	15	1	15				15						15	15
4.	1	1	1	1	5	5	5	1		1	5	1	1	1	15	15	5	1	15	1	15	1	15	25	25	25	15	25	25			25	15	15
5.	1	1	1	1	5	5	5	1	9	1	5	1	1	1	15	15	5	1	15	1	15	1	15	25	25	25	15	25	25	9	9	25	15	15
6.	1	1	1	1	5	5	5	1	15	1	5	1	1	1	15	15	5	1	15	1	15	1	15	25	25	25	15	25	25	15	15	25	15	15

1. step: Creating community 5 of parent 1.
2. step: Creating community 15 of parent 1.
3. step: Community 7 of parent 2 cannot be created, community 1 of parent 2 is created.
4. step: Creating community 25 of parent 1.
5. step: Creating community 9 of parent 2 and then stop because all nodes are assigned to community.
6. step: After merging community 9 with community 15 the modularity is 0.4198.

Algorithm 1: Algorithm NE-Net.

Data: Graph $G = (V, E)$, population size N, mutation probability p_m, crossover probability p_c, maximum number of generations $g_{\max}$
Result: Partition X^* with a maximal modularity.
1. Initialization:
$P_0 = \{X_1, X_2, ..., X_N\} \leftarrow$ Initialize Population(G,N)
2. values of criteria function for all individuals are calculated.
$t \leftarrow 0$**repeat**
  $t \leftarrow t + 1$;
  3. The hybridized genetic operators mutation and crossover procedures 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_{t-1})
  if rand(0..1) $> p_c$ then $P^o \leftarrow$ Q/n-crossover operator based on the node entropy (P^o)
  if rand(0..1) $> p_m$ then $P^o \leftarrow$ mutation operator(P^o)
  4. Repair individuals in offspring population in the following two steps:
  4.1 Merge communities to increase the modularity.
  4.2 Reasign each node to the community with the lowest values of node entropy.
  5. $Pt \leftarrow$ Keep the N best individuals in the population $Pt + P^o$.
  Identify $X_{\max Q}$
  Go to step 2 until the iteration number reaches $g_{\max}$.
until $t < g_{\max}$;
return $X_{\max Q}$

is inserted into the population and the best individuals are selected and included in a new population. The algorithm stops when the number of generations reaches the given threshold. The detailed process is described as follows.

Step 1: Initialization. Population initialization is the first task in evolution algorithms. It is important, because it affects the quality of the solution and also the convergence speed. Since no information about the solution is usually available, random initialization of genes of chromosomes is used to generate candidate solutions (initial population). Initially, each gene representing graph node is put in a different community for all individuals in the initial population. We use gene indices ($x_p^i = i; i = 1, \ldots n$) for community labels. Then for all chromosomes for each gene i of n genes, the algorithm selects randomly one neighbor node k for each node i in the graph and set $x_p^i = k$. We choose one neighbor completely randomly which provides generating unsupervised solutions, although some research results prove that maximal neighbor similarity reveals real communities in networks [28]. So each gene contains the index $(1, \ldots n)$ of one of its neighbors. Then each gene of each individual is randomly chosen and assigned to the same community as the gene with index with the label x_i^j: $(for\ (\forall x_i^j\ and\ random\ j \in [1 \ldots n])\ x_i^j \leftarrow x_i^{x_i^j})$. The generated solutions are unsupervised and from the possible solution space, while the proposed initialization process takes into account only the effective connections of the network.

Step 2: Values of the modularity function of each chromosome of the initial population P_0 are calculated.

Step 3: New iteration t ($t = 1, 2, \ldots, g_{\max}$) starts, where $g_{\max}$ is number of generations. Proportional cloning to population P_{t-1} forms the first offspring population. Then genetic operators refine the offspring population and introduce new genetic material to the offspring population. After binary tournament selection, mutation and modularity crossover operators are performed on the clone population to create child chromosomes of the offspring population P^o.

Mutation allows a new genetic material to enter into the population with changing part of existing individuals randomly. Mutation should not occur often because it causes a random search. Individuals with lot of mutated genes usually do not survive the selection. We use a neighbor-based mutation operation which takes into account only the effective connections [25]. The mutation on a gene is performed if a generated random value $r \in [0, 1]$ is smaller than the mutation probability p_m. The value (community label) of this gene is replaced with one of its neighbors' label: $(x_a^i \leftarrow x_a^j,\ \exists j \in \{j | A_{ij} = 1\})$.

Crossover inherits the best properties into new population. We use crossover to inherit the best communities with the highest ratio of modularity and number of nodes in a community from one generation to the next. Two parents for a crossover operation are selected with a tournament selection of chromosomes from current population. All communities from both parents are sorted by the ratio of modularity and number of nodes assigned to the community. First we create the community in the offspring that is equal to the community with the highest ratio and then with the second-best ratio and so on until all nodes in the offspring are assigned to one community (as shown in Section 3.1.1).

Step 4: The child chromosome is modified in the following two steps.

Step 4.1: Communities can be merged to increase modularity. Using modularity based crossover more small communities with only a few or even a single node can be generated in chromosomes of the offspring population P^o. Modularity can be increased also by merging of single nodes or more small communities with real communities. For merging multiple pairs of communities at a time, we use the same criteria as Multi-Step Greedy agglomerative algorithm (MSG) [30]. At each iteration

the modularity is calculated after merge of each pair of connected communities. Two communities are joined when the modularity change is the greatest and positive and neither the first community nor the second community do not cause the highest modularity change with any other community.

Step 4.2: Nodes are reassigned. In each generation, a local search procedure is applied to individuals in the population to obtain better individuals. Each node on the border of each community can be moved to one neighbor community. Each node that is not allocated to a community with the greatest part of the node entropy are moved to this community in this step. So we improve the quality of communities of the offspring population P^o in this step by satisfying node entropy constraint as discussed in the Section 2.1.

Step 5: Model selection. First, a combined population is formed $R_t = P_t + P^o$. Each solution is assigned a rank (1 is the best level, 2 is the next-best level, and so on) equal to its nondomination level. Then, the population R_t is sorted according to nondomination. Crowding distance values of all nondominated individuals are calculated and the first N individuals are chosen as a new population. The dominant population is updated. Repeat all steps from Step 2 until the iteration number reaches $g_{\max}$.

3.3. Partition entropy and algorithm E-Net

We introduce and use the partition entropy instead of modularity function to avoid resolution limit and identify all significant well-separated communities regardless of the community sizes. The partition entropy is based on the concept of Shannon entropy to measure a network's information. We consider the network with all nodes and edges maximal information about network and the partition of network into communities an abstraction of network, while we consider a community detection process as a process of information loss.

Node entropy used for local learning is measurement of uncertainty about the node and natural community. Its maximum appears when all communities of neighbors are equally similar to a node. When we go away from equally assignment of a node to communities the entropy decreases. When each node belongs to the real community the entropy is low and the real community structure of a network is uncovered.

Partition entropy measures disorder of each community. The more similar the elements of community, more ordered is community and less entropy has community. The smaller difference between internal and external edges of community per number of nodes of community, the more similar are nodes of community and more separated from their surrounding and less entropy has community.

$$entropy_C = \begin{cases} n_C * (-sim_C * \log(sim_C)) & when\ sim_C > 0 \\ n_C * (1.0 - (-sim_C) * \log(-sim_C)) & when\ sim_C < 0 \end{cases} \quad (6)$$

where n_C is number of nodes in community C and similarity of nodes of community sim_C.

$sim_C = \frac{In_C - Out_C}{2*(In_C + Out_C)}$; $In_C = \sum_{i,j \in C}(A_{ij})$; $Out_C = \sum_{i \in C, j \notin C}(A_{ij})$

In_C and Out_C are numbers of internal and external edges of community C respectively. Partition entropy $entropy_P$ measure tries to find a partition P with low entropy of communities and keeping in mind the modularity of partition.

$$entropy_P = \frac{\sum_{i=1}^{k} entropy_{C_i}}{k} \quad (7)$$

where k is number of communities of partition P.

The time complexity of the proposed algorithm is analyzed. For a network with n nodes and m edges, we first have to initialize population. This require $O(N*n)$ time, where N is the size of population. At each iteration we need to perform the crossover operator $N/2$ times and the mutation operator N times at worst. The time complexity of the calculation of modularity is $O(m)$. Genetic operators require $O(N(n+m))$. The time complexity of the learning strategy (step 4. repairing individuals in offspring population) is $O(h*m)$, where h is the number of steps, which is smaller than $\log n$. Merging pairs of communities requires $O(m*\log n)$ and the number of iterations is $\log n$, so the whole time complexity is $O(m(\log n)^2)$ [30]. The time complexity for updating a population depends on the size of the population N and size of the offspring population N_o. At most $O((N+N_o)^2)$ comparisons are required to identify nondominated individuals in combined population. The worst time complexity for sorting individuals by crowding distance is $O((N+N_o)\log(N+N_o))$. In real applications $m > n, n > N + N_o$, so the time complexity of the algorithm is $O(m * (\log n)^2)$. This time complexity is comparable to memetic community detection methods used in experiment described in the next section, but is higher to the time complexity of single objective Louvian method, which is not genetic algorithm.

4. Experimental results

In this section, we discuss results of performed experiments on some artificial and some real-world networks, that are often used for evaluation of new methods.

Modularity is used as a quality measure for evaluation of the obtained partitions of real-world networks by NE-Net.

Partition entropy is used to avoid resolution limit and to identify also small communities by E-Net. Obtained results proved that partitions obtained by E-Net have more communities and contain also small communities.

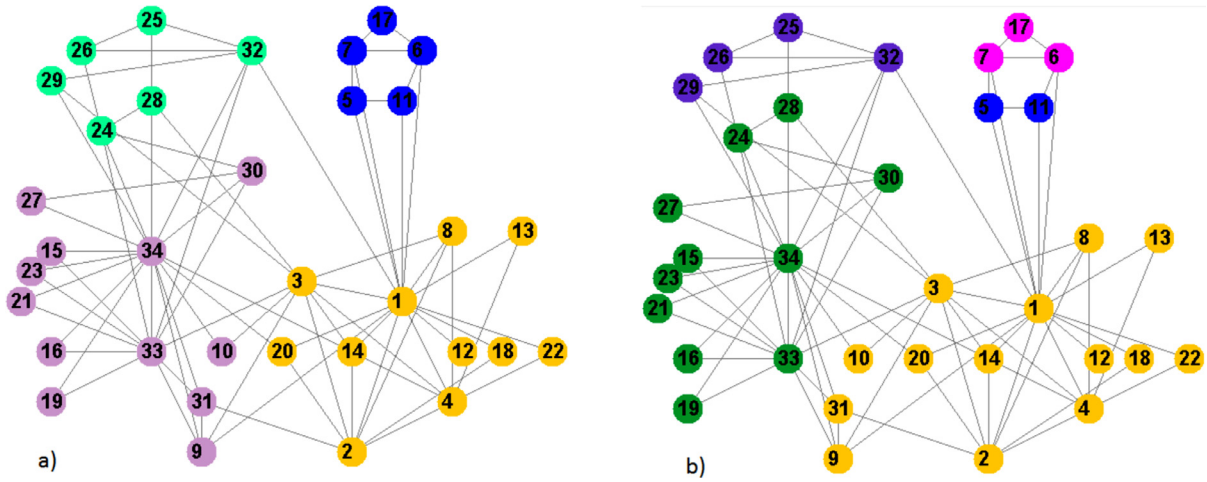

Fig. 2. Zachary Karate Club network: (a) 4 uncovered communities by NE-Net, (b) 5 uncovered communities by E-Net.

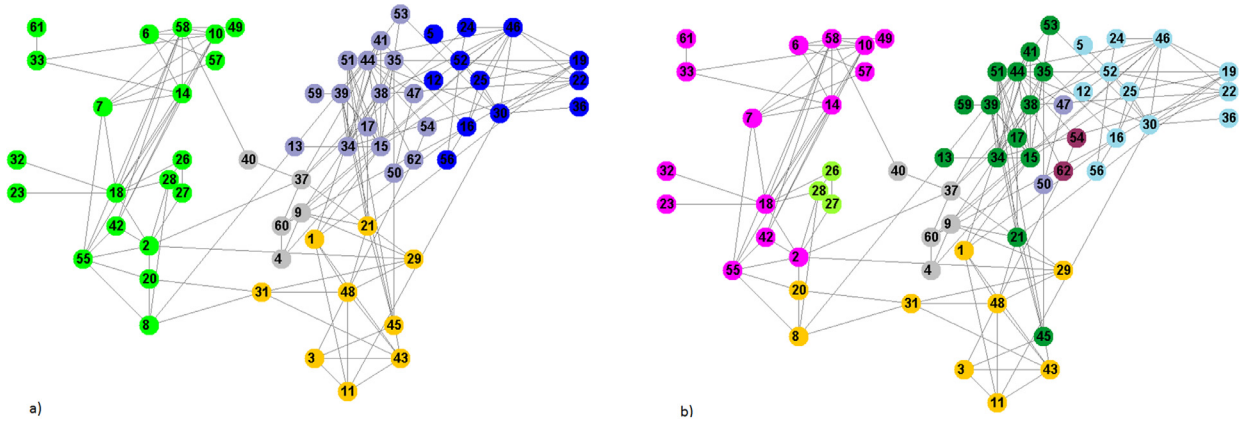

Fig. 3. Dolphin network: (a) 4 communities discovered by NE-Net, (b) 8 communities uncovered by E-Net.

We performed experiments on real-world and synthetic networks to illustrate that the proposed NE-Net method can find natural partitions effectively. The obtained results are also compared to the best single objective community detection method Louvain and to single objective algorithm Meme-Net and to two multi-objective genetic algorithms MOGA-Net and MOCD.

4.1. Experiments with some typical real-world networks

4.1.1. Zachary Karate Club network

Zachary's Karate Club is a well-known social network consisting of 34 members of a Karate Club and 78 pairwise links between members who interacted outside the club [37]. 4 communities are identified in the resulted partition with modularity 0.4198 shown in Fig 2a, while E-Net uncovered 5 communities shown in Fig 2b.

4.1.2. Dolphin network

Dolphin network is an undirected social network of frequent communications between 62 dolphins in a community living in New Zealand [20]. 5 communities are identified in the resulted partition with modularity 0.5285 shown in Fig. 3a.

4.1.3. College football network

Network of American football games between Division IA colleges during a regular season Fall 2000 [11]. The teams are divided into 12 conferences with intra-conference games being more frequent than inter-conference games. 10 communities are identified in the resulted partition with modularity 0.6058 shown in Fig. 4a, while 11 communities identified by E-Net shown in Fig. 4b.

Jazz musicians network [12] is the network of Jazz musicians. Dataset Neural represents the neural network of *C. elegans* [36]. Metabolic network dataset lists edges of the metabolic network of *C. elegans* [7]. Email network dataset contains e-mail

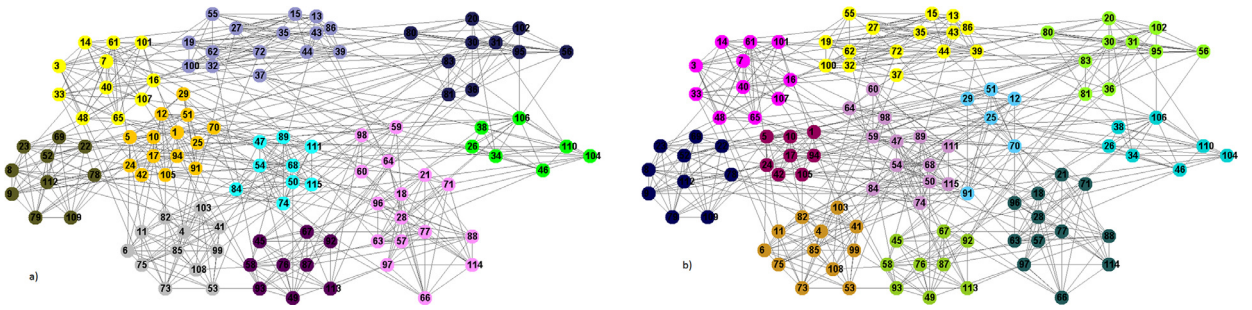

Fig. 4. Football network: (a) 10 communities marked with different colors identified in the Football network by NE-Net and (b) 11 communities identified by E-Net.

Table 3
The maximal modularity, average modularity and standard deviation of modularity obtained by NE-Net and evolutionary method using a crossover based on the ratio of modularity and number of elements compared to the obtained values of published methods Louvian, MOCD, MOGA-Net and MEME-Net.

Network	Q	Louvian	MOCD	MOGA-Net	MEME-Net	NE-Net
karate	$Q_{\max}$	0.418	0.416	0.416	**0.42**	**0.42**
	Q_{avg}	0.418	0.416	0.395	0.419	**0.42**
	Q_{std}	–	**0**	0.008	0.002	**0**
dolphins	$Q_{\max}$	0.519	**0.529**	0.525	0.519	**0.529**
	Q_{avg}	0.519	0.524	0.528	0.51	**0.529**
	Q_{std}	–	0.003	0.007	0.006	**0**
polbooks	$Q_{\max}$	0.520	**0.527**	0.499	0.523	**0.527**
	Q_{avg}	0.520	**0.526**	0.463	0.522	**0.526**
	Q_{std}	–	**0.001**	0.012	0.003	**0.001**
football	$Q_{\max}$	0.604	0.602	0.433	0.604	**0.606**
	Q_{avg}	0.604	0.587	0.396	0.602	**0.606**
	Q_{std}	–	0.007	0.017	0.002	**0.001**
jazz	$Q_{\max}$	0.445	0.437	0.293	0.442	**0.443**
	Q_{avg}	0.445	0.428	0.293	0.442	**0.442**
	Q_{std}	–	0.005	0.008	**0.0001**	0.004
Neural	$Q_{\max}$	0.383	0.374	0.243	0.379	**0.399**
	Q_{avg}	0.383	0.344	0.137	0.31	**0.396**
	Q_{std}	–	0.017	0.069	0.057	**0.015**
Metabolic	$Q_{\max}$	**0.432**	0.393	0.273	–	**0.433**
	Q_{avg}	**0.432**	0.384	0.221	–	0.408
	Q_{std}	–	**0.006**	0.027	–	0.025
email	$Q_{\max}$	**0.572**	0.477	0.346	–	0.526
	Q_{avg}	**0.572**	0.443	0.307	–	0.521
	Q_{std}	–	0.022	0.022	–	0.022
Netscience	$Q_{\max}$	**0.959**	0.941	–	–	**0.959**
	Q_{avg}	**0.959**	0.927	–	–	0.951
	Q_{std}	–	0.007	–	–	**0.005**

interchanges between members of the Univeristy Rovira i Virgili [14]. Netscience network records coauthorship of scientists working on network theory and experiments [22].

4.2. Comparison with other algorithms

The performance of the proposed method is compared to the best single objective community detection method Louvain and to single objective algorithm Meme-Net and to two multi-objective genetic algorithms MOGA-Net and MOCD on GN benchmark and 9 real-world data set. All the non-deterministic algorithms have been independently run 10 times on each data set. The results of E-Net and NE-Net and compared methods on the considered nine real-world data sets in 10 runs is shown in Table 3. Standard deviations of deterministic algorithm Louvian is removed. Bold numbers in each row denote the best values. The parameters of NE-Net and E-Net are the following: population size 32, maximum generation number $g_{\max} = 5$, crossover probability p_c=0.8, mutation probability p_m=0.2. Modularity Q is used as the criterion to measure the quality of the obtained partitions, because modularity is the most often used quality function whose higher value indicates stronger community structure. $Q_{\max}$ and Q_{avg} obtained by NE-Net on most real-world networks are higher than those of

Table 4
Results of the modularity analysis on real networks. The first raw contains the number of modules detected in the partition obtained for the maximal modularity and the second row contains the total number of submodules and the corresponding value of the modularity of the partition, which is lower than the peak modularity initially found (reported in [9]). In the third and fourth row, we report the number of communities and Q obtained by NE-Net and E-Net for Yeast, *E. coli*, Electric circuit and *C. elegans* dataset.

Network nodes edges	Yeast 688 1079	*E. coli* 423 519	Electric circuit 512 819	*C. elegans* 306 2345
No. communities for max Q (Q_{max})	9 (0.7396)	27 (0.7519)	11 (0.6701)	4 (0.4022)
Max. no. communities (Q)	57 (0.6770)	76 (0.6615)	70 (0.6401)	15 (0.3613)
NE-Net: No communities (Q)	45 (0.778)	45 (0.7792)	14 (0.8535)	7 (0.3892)
E-Net: No communities (Q)	50 (0.694)	62 (0.7074)	46 (0.7019)	8 (0.3755)

single objective optimization method Louvian and single objective memetic algorithm Meme-Net and multi-objective genetic algorithms MOGA-Net and MOCD with only a few exceptions.

We compared results obtained by E-Net and NE-Net with the results reported in the research of resolution limit Ref[9]. on the following 4 networks: Escherichia coli, yeast, electronic circuits (all from Ref. [15].) and Caenorhabditis Elegans [16]. In Table 4 the number of nodes, edges and resulted number of communities obtained by NE-Net and E-Net and the corresponding value of the modularity of obtained partitions are given. In the first and second row of the Table 4, we report the number of modules detected in the modularity maximum Q_{max} by using simulated annealing and the reported total number of submodules and the corresponding value of the modularity of the partition, found in the research of resolution limit Ref[9]. Maximum modularity of all these networks is very high, with values from 0.40 (*C. elegans*) to 0.75 (*E. coli*). As shown in Table 4, the E-Net identified more communities (also small) in all considered networks than NE-Net and more than is the number of uncovered communities with maximal modularity Q. The modularity of partitions identified by E-Net is lower than by NE-Net although they have more communities than partitions identified by NE-Net. An increase of the number of modules does not necessarily correspond to an increase in modularity because the modules would be smaller and so would be each term of the sum. The number of identified communities obtained by E-Net is lower and modularity of partitions is higher than in the modularity maximum partitions obtained by using simulated annealing in the research of resolution limit. We have verified that all communities detected by E-Net are indeed modules, i.e. they satisfy the modularity (Eq. 2) of community greater than 0.

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. We use the quality measurement Normalized Mutual Information NMI [3], defined by Eq. 8.

$$NMI(R,P) = \frac{-2\sum_{r=1}^{N_R}\sum_{p=1}^{N_P} D_{rp}\log\left(\frac{D_{rp}N}{D_{rs}D_{sp}}\right)}{\sum_{r=1}^{N_R} D_{rs}\log\left(\frac{D_{rs}}{N}\right)+\sum_{p=1}^{N_P} D_{sp}\log\left(\frac{D_{sp}}{N}\right)} \tag{8}$$

where R is partition of network with N elements into N_R real communities and P is partition of the same network into N_P predicted communities. D is matrix with elements D_{rp} counting the number of nodes in community r of R that also appear in community p of P. D_{rs} is the sum over row r of D while Dsp is the sum of elements in column p.

We have evaluated the accuracy of the proposed algorithm on GN artificial networks [11]. Each GN network consists of 128 nodes forming four communities with 32 nodes each. All nodes have the same average degree (number of neighbors) 16 and each node has (1 - μ) internal edges with the nodes of its community, where μ is the mixing parameter. For μ greater than 0.5, the average number of neighbors of each node in its community is smaller than that of its outside neighbors, the network consists of weak-separated communities which form partition with weak community structure. And weak-separated communities are very difficult to uncover. The comparisons of average values of NMI obtained by NE-Net with reported values of NMI for two other compared methods are shown in Fig. 5. NE-Net has a good performance for uncovering communities of artificial GN networks.

5. Conclusion

In this paper, the automatic network community detection is formulated as an optimization problem facilitated by node entropy. Based on this formulation the evolutionary algorithm maximizing the modularity criterion function is proposed where use of the node entropy accelerates convergence to an optimum. Furthermore, with using node entropy and problem-specific genetic operators, the accuracy of the community detection increases. The efficiency of the proposed algorithm is verified by comparative experiments on real world data sets and synthetic GN benchmark networks.

Moreover, this algorithm is slightly modified in order to avoid modularity function which suffers a resolution limit and to make the proposed algorithm able also to identify small communities in cases where communities are unambiguously defined. Partition entropy is used instead of modularity function. The results for real networks show that partition entropy helps to detect the ground-truth community structure more accurately than using modularity. There are many possible

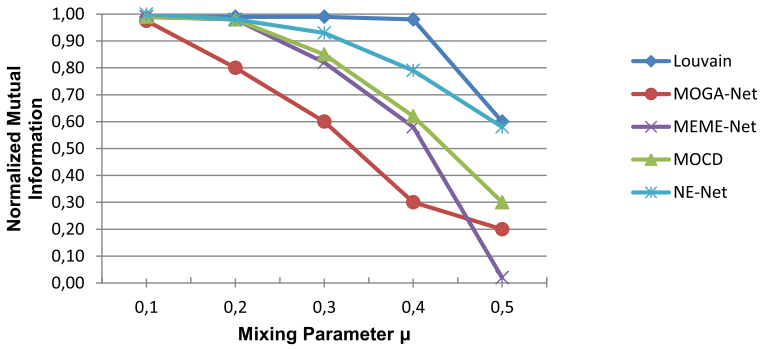

Fig. 5. Performance comparisons between Louvain, NE-Net, Meme-Net and MOGA-Net and MOCD on the GN benchmark networks with different mixing parameters μ ranging from 0.1 to 0.5.

directions for future work. NE-Net can be extended with incorporating other sources of information that influence node entropy like edge attributes.

Acknowledgment

This work was supported by the Slovenian Research Agency (grant no. P2-0041, J2-8176).

References

[1] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, D. Wagner, On modularity clustering, IEEE Trans. Knowl. Data Eng. 20 (2) (2008). 172/18.
[2] M. Chen, K. Kuzmin, B.K. Szymanski, Community detection via maximization of modularity and its variants, IEEE Trans. Comput. Soc. Syst. 1 (1) (2004) 46–65.
[3] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. Theory Exp. 9 (2005) P09008.
[4] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, Chichester, 2001.
[5] K. Deb, A. Pratap, S.A. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[6] S.N. Dorogovtsev, J.F.F. Mendes, Evolutionary of Networks: From Biological Nets to the Internet and WWW, Oxford University Press, 2003.
[7] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2005) 027104.
[8] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–114.
[9] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. USA 104 (1) (2007) 36–41.
[10] M.R. Garey, D.S. Johson, Computers and Intractability: A Guide to the Theory of NPCompleteness, W.H. Freeman, 1979.
[11] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (2002) 7821–7826.
[12] P.M. Gleiser, L. Danon, Community structure in jazz, Adv. Complex Syst. 06 (2003) 565–573.
[13] M. Gong, B. Fu, L. Jiao, H. Du, A memetic algorithm for community detection in networks, Phys. Rev. E 84 (5) (2011) 056101.
[14] R. Guimerá, L. Danon, A. Diaz-Guilera, F. Giralt, A. Arenas, Self-similar community structure in a network of human interactions, Phys. Rev. E 68 (2003) 065103.
[15] http://www.weizmann.ac.il/mcb/UriAlon/supplementary-material-milo-et-al-science-2002.
[16] http://toreopsahl.com/datasets/.
[17] S. Jianhai, T.C. Havens, Quadratic program-based modularity maximization for fuzzy community detection in social networks, IEEE Trans. Fuzzy Syst. 23 (5) (2015) 1356–1371.
[18] J.N. Kapur, H.K. Kesavan, Entropy Optimization Principles with Applications, Academic Press, San Diego, 1992.
[19] A. Lancichinetti, S. Fortunato, Limits of modularity maximization in community detection, Phys. Rev. E 84 (6) (2011).
[20] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. 54 (4) (2003) 396–405.
[21] P. Moscato, On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms, in: Caltech Concurrent Computation Program Report, 1989, p. 826.
[22] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (2006) 036104.
[23] Y. Park, M. Song, A genetic algorithm for clustering problems, in: Proceedings of the Third Annual Conference on Genetic Programming, 1998, p. 568-657.
[24] C. Pizzuti, Ga-net: a genetic algorithm for community detection in social networks, in: PPSN, 2008, p. 1081-1090.
[25] C. Pizzuti, A multiobjective genetic algorithm to find communities in complex networks, IEEE Trans. Evol. Comput. 16 (2012) 418–430.
[26] A. Pothen, H. Sinmon, K.P. Liou, Partitioning sparse matrices with eigenvectors of graphs, SIAM J. Matrix Anal. Appl. l (11) (1990) 430–452.
[27] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying clusters in networks, Proc. Natl. Acad. Sci. USA 101 (9) (2004) 2658-.
[28] K.R. Žalik, Maximal neighbor similarity reveals real communities in networks, Sci. Rep. 5 (2015) 1837.
[29] K.R. Žalik, B. Žalik, Multi-objective evolutionary algorithm using problem-specific genetic operators for community detection in networks, Neural Comput. Appl. (2017) 1–14, doi:10.1007/s00521-017-2884-0.
[30] P. Schuetz, A. Caflish, Efficient modularity optimization by multistep greedy algorithm and node refinement, Phys. Rev. E 77 (4) (2008) 046112.
[31] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (3) (1994) 221–248.
[32] C.E. Shannon, W. Weaver, The Mathematical Theory of Communication, University of Illinois Press, 1949. ISBN 0-252-72548-4
[33] C. Shi, Z. Yan, Y. Cai, B. Wu, Multi-objective community detection in complex networks, Appl. Soft Comput. 12 (2012) 850–859.
[34] S.H. Strogatz, Exploring complex networks, Nature 410 (2001) 268–276.
[35] M. Tasgin, A. Herdagdelen, H. Bingol, Community detection in complex networks using genetic algorithms, 2007. ArXiv: 0711.0491.
[36] D.J. Watts, S. H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1998) 440–442.
[37] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (1977) 452–473.