MULTI-OBJECTIVE COMMUNITY DETECTION METHOD USING AN IMPROVED NSGA II ALGORITHM

Krista Rizman Žalik
University of Maribor, Faculty of electrical engineering and computer science, Maribor, Slovenia
Email: krista.zalik@um.si

ABSTRACT
Community detection methods provide help in understanding the structure and function of networks. Since single objective community detection methods are not able to detect multiple significant community structures, some methods formulate the community detection as multi-objective problems. This paper proposes a multi-objective approach using a combination of three objective functions in multi-objective framework that is able to discover wide specter of resulting partitions. The algorithm is based on the basic NSGA-II, and improvements are as follows. A hybrid model is established for population initialization and the mutation operator and the crossover operators compose only possible partitions of input network. Different three optimization functions are tested in order to detect a large spectrum of resulting partitions.

KEY WORDS
Complex networks, community detection, multi-objective optimization, evolutionary algorithms

1. Introduction

There exist various kinds of complex systems which can be represented as complex networks, such as social networks [1] or the Internet [2]. Community detection is important while communities describe the structure and functional relationship. Uncovered communities identify structures, which correspond to important functions. For example, the modular structure WWW is a set of web pages sharing the same topic [3].

Communities are groups of nodes that are densely interconnected but only sparsely connected with the rest of the network and different community detection methods have been proposed [4].

1.1 Community Detection as Single-objective Problem

In order to extract similar groups of nodes, one objective function is chosen that creates a community as a group of nodes with higher internal connectivity than external connectivity. The objective is usually NP-hard to optimize, so approximation algorithms such as genetic algorithms [5] or heuristics are usually used (e.g., between [6] and different optimization functions [7]) to find partitions of nodes that approximately optimize and corresponds to real communities. Modularity optimization is often used function although it has a resolution limit [8]. Resolution limit means that community detection methods cannot identify well separated communities smaller than a scale. The size of the smallest identified community depends on the total size of the network and on the degree of inter-connectedness of the modules.

The conventional community detection is a single-objective optimization problem since the community detection corresponds to discover a community structure that is optimal on one single-objective function. These single-objective approaches have been successfully applied to both artificial and real problems. However, the single-objective community detection returns one solution with a particular community structure property defined by the used objective function. In addition, different objective functions result in different partitions. Some objective functions generate a huge number of small communities, while the others return a small number of very big communities. And networks can have more partitions describing multiple structures such as hierarchical or overlapping. Three optimization functions are used in order to detect a large spectrum of the solution partitions.

1.2 Community Detection as Multi-objective Problem

In this paper, we take the community detection as a multi-objective optimization problem [9]. The advantage of multi-objective community detection algorithms is more discovered community structures optimal on multiple objective functions.

In order to effectively solve this problem, we propose a multi-objective community detection algorithm that returns a set of solutions. These solutions can reveal the implicit structure information such as overlapping or hierarchical. The solution partitions are optimized by the three optimization functions.
2. Multi-objective Optimization Definition

A multi-objective optimization problem is a problem of minimizing of multiple optimization functions. There are multiple conflicting objectives in community detection. Therefore there are more solutions and more best solutions and not only one identified in single objective optimization problem. Each solution describes different partition with different number of communities. The number of communities is automatically determined by the optimal tuning of values of both optimization functions. The taken objective functions have to be conflicting and increasing one have to cause decreasing the value of the other objective function. The result of multi-objective optimization is a number of Pareto optimal solutions. A solution is called non-dominated or Pareto optimal, if none of the objective functions can decrease the value without increasing the other objective values in the case of minimizing objective functions. Pareto optimal solutions are equally good. Some additional information is necessary to select one or some of the solutions to be the best.

The definition of dominance helps to identify different best solutions: A solution x1 dominates another solution x2, if the solution x1 is no worse than x2 in all objectives and the solution x1 is strictly better than x2 in at least one.

A multi-objective optimization problem is a problem of minimizing of multiple optimization functions. There are multiple conflicting objectives in community detection. Therefore there are more solutions and more best solutions and not only one identified in single objective optimization problem. Each solution describes different partition with different number of communities. The number of communities is automatically determined by the optimal tuning of values of both optimization functions. The taken objective functions have to be conflicting and increasing one have to cause decreasing the value of the other objective function. The result of multi-objective optimization is a number of Pareto optimal solutions. A solution is called non-dominated or Pareto optimal, if none of the objective functions can decrease the value without increasing the other objective values in the case of minimizing objective functions. Pareto optimal solutions are equally good. Some additional information is necessary to select one or some of the solutions to be the best.

The definition of dominance helps to identify different best solutions: A solution x1 dominates another solution x2, if the solution x1 is no worse than x2 in all objectives and the solution x1 is strictly better than x2 in at least one.

2.1 Related Work

There have been already proposed some multi-objective detection methods. Multi-objective community detection algorithm (MOCD) [10] uses both terms of modularity measure introduced by Newman [11] as two objective functions. As the main frameworks of MOCD algorithm the Pareto Envelope-based Selection Algorithm version 2 (PESA-II) [12] is used.

The other proposed multi-objective approach, named MODA-Net [13] optimizes two objective functions introduced in [14] and [15]. The first objective function measures community score. The high community score denotes dense communities. The second used objective function is node that is named also community fitness. When both objective functions have its maximum value, the number of external links between communities is minimized.

The multi-objective evolutionary algorithm based on decomposition has been proposed [16].

In [17] the community detection was formulated as a multi-objective optimization problem which regards the two contradictory parts of modularity function as two objective functions. In order to optimize such two objective functions effectively, a multi-objective memetic algorithm named as MMCD has been proposed.

3. Objective Measures

3.1 Objective Functions

Many objective functions have been proposed and used for community detection [18]. In community detection objective functions are used to capture a group of nodes with better internal connectivity than external. We summarize some objective functions used for community detection.

Cut-based criteria:
- Conductance is the fraction of total edges that points outside the cluster [19].
- Expansion is the number of edges per node that points outside the cluster [20].
- Cut ratio is the fraction of all possible edges leaving the cluster [21].

Degree based criteria:
- Maximum out degree fraction (ODF) is the maximum fraction of edges of a node pointing outside the cluster [22].
- Flake out degree fraction (ODF) is the fraction of nodes in S that have fewer edges pointing inside than to the outside of the cluster [22].

Some other criteria:
- Modularity Q measures the number of within-community edges, relative to a model of a random graph with the same node degree [11]. It has to be maximized.
- Internal density is the internal edge density of the cluster [19].
3.2 Comparison of Objective Functions

Only Community score and Internal density have no resolution limit and community score always find the correct partition, while internal density divides clique into smaller communities and over-divides the network was reported by Shi et al. [23]. They explored many objective functions and reported that different optimization functions can identify different quality of communities. Conductance means the ratio of edges going outside to nodes of other communities is smaller. Conductance means sparsely connected communities with the outside while shortest path means densely interconnected. Shorter path is the average shortest path of nodes in the community. A smaller shortest path shows that the nodes in the community are densely connect with each other. The cut based criteria and modularity Q have smaller shorter path, while description length have larger shortest path for large communities. Some optimization functions favors to identify only well separated communities while the others identify small number of huge communities. Cut based criteria and internal density generate some huge communities with small conductance. Huge communities have some edges connecting with outside and more edges inside. Maximum number of communities can be found by community score and the modularity and the cut based criteria reveal the minimum number of communities. Optimization functions identify also communities of different size. The cut based criteria always find larger communities, community score and modularity have the opposite trend. Cut based criteria and description length tend to find small number of networks with large size. Optimization functions identify communities of different granularity. The cut-based criteria, degree based criteria and description length tend to divide the networks with coarser granularity. Community size is large and the number is small. Opposite to them community score and modularity identify large number of communities with small size. Internal density is between and can divide the networks into some large communities as well as many small communities.

4. Used Objective Functions

Many objective functions have been proposed over the years and there is still open question which are the most appropriate functions to be used together to solve the community detection problem as multi-objective problem we used the following 6 functions.

Let $S$ be a set of nodes in community $C$, $n_S$ is a number of nodes in the set $S$ and $n$ is the number of all nodes in the graph $G$, $e^\text{out}_S$ is the number of links that go to other communities and $e^\text{in}_S$ is the number of internal links. Cut ratio is the fraction of all possible edges leaving the community.

$$F1(P) = \sum_{S \in P} \frac{e^\text{out}_S}{n_S(n_S-1)},$$

Cut ratio favors to form small number of large communities, so we used the other objective function - internal density that favors the partitions with large number of small communities.

Internal density:

$$F2(P) = \sum_{S \in P} 1 - \frac{2e^\text{in}_S}{n_S^2}$$

A community should contain nodes that have fewer edges pointing inside than to the outside of the cluster. Flake-ODF optimization function measures this:

$$F3(P) = \sum_{S \in P} \left| e'^i_{jS} - e'^j_{iS} \right| \frac{deg_i}{n_S}$$

where $deg_i$ is the degree of vertex $i$ and $e'^i_{jS}$ is an outside edge of a node $i$ to a node $j$ that belongs to different community than node $i$.

We also used modularity $Q[30]$. Modularity of $k$ communities, where $l_S$ is the number of internal edges of community $S$ and $d_S$ is sum of degrees of nodes and $m$ is number of edges:

$$F4(P) = 1 - Q = 1 - \sum_{i=1}^k \frac{l_S}{m} - \frac{d_S^2}{4 \cdot m^2}$$

We also tested with an optimization function that minimizes differences on the border $F5(P)$ and optimization function that only maximizes number of communities $F6(P)$.

$$F5(P) = \sum_{S=1}^k (e^\text{in}_S - e^\text{out}_S) \cdot \frac{n_S}{n}; \quad F6(P) = \frac{1}{k}$$

5. Algorithm

In this section we give a description of the proposed multi-objective algorithm for community identification in the network that uses three objective functions for uncovering wide spectrum of resulting partitions containing different number of communities of different sizes.

In the last few years efforts have been given in evolutionary computation to the development of multi-objective optimization algorithms. The Multi-objective Genetic Algorithm we used as a framework in the proposed methods is the Non-dominated Sorting Genetic Algorithm (NSGA-II) proposed by Srinivas and Deb in [24],[25]. NSGA-II builds a population of competing
individuals and ranks them on the basis of non-dominance.

The algorithm NSGA-II has been adapted with a customized population type that suitably represents a partitioning of a network and three objective functions.

![Gene to node representation](image)

### 5.1 Genetic Representation

In this paper we adopt the locus-based adjacency representation (LAR) proposed by [26]. In this representation, any individual chromosome in the population consists of \( n \) genes. Each gene corresponds to a node in network with \( n \) nodes. Each gene \( i \) can have an arbitrary allele value \( j \) from the set \( \{1, \ldots, n\} \). Gene \( i \) and its allele value \( j \) denote that nodes \( i \) and \( j \) are in the same community. At the beginning allele values of each node are initialized to have a value of one of its neighbor node. So when the evolution process of finding communities is finished all communities have to be decoded so that each pair of connected nodes \( i \) and \( j \) denoted by the allele value \( j \) of the \( i \)-th gene, are in the same community. Fig. 1 shows an example network and its locus-based adjacency representation by two possible corresponding genotypes. The last one is obtained after the finish of evaluation and decoding processes. All nodes with the same values of alleles of corresponding genes belong to the same community.

### 5.2 Initialization

Initialization considers all edges of the graph. Two thirds of individuals are generated randomly. In the \( i \)-th place can be allele value \( j \) if there is a connection between nodes \( i \) and \( j \) in the graph. So the initial population contains only possible solutions. In addition, we speed the community detection process by initializing one third of individuals with starting values that are close to possible solutions. For each gene \( i \) one of the 3-most similar neighbor nodes of node \( i \) is assigned randomly as allele value in the \( i \)-th place of individual genotype. Number of common neighbors is used as a similarity measure.

### 5.3 Uniform Crossover

Uniform crossover generates child individual from two parents of population. First, the random binary vector of size \( n \) (\( n \) is number of nodes in the graph \( G \)) is created. When the value in the \( i \)-th place is 0 the allele value \( j \) of the first parent is chosen for the \( i \)-th gene, while for value 1 the allele value of the \( i \)-th gene from the second parent is chosen. Because for both parent pairs \( i \) and \( j \) there is an edge between nodes \( i \) and \( j \) in the graph, this is true also for the generated child individuals. The allele value of the \( i \)-th gene is from one of two parents.

### 5.4 Mutation

The value \( j \) of the \( i \)-th gene is randomly changed to label of one neighbor of the node \( i \). So the mutated individual represents one possible solution.

For a given graph \( G \) that models network \( N \), the proposed algorithm starts with initialized population consisting of individuals and each of them can be possible solution. Then the algorithm generates new populations for the fixed number of iterations. First, the algorithm computes values of objective functions, assigns the rank to each individual according to Pareto dominance set and sorts them. New population is generated using the crossover and mutation operators. The resulted set of partitions contained in the Pareto front has different number of communities and represent different results of balance between all used optimization functions. So the whole hierarchy of solution can be observed. One quality measure has to be used to identify the best solution.
6. Experimental Results

6.1 Method Evaluation

We compared the results obtained by the proposed algorithm with GN algorithm (http://cs.unm.edu/~aaron/research/fastmodularity.htm) on some well-known real world data networks from the literature with the known community structure that are often used for testing of community detection methods. We used standard parameters for genetic algorithm mutation rate 10% and crossover rate 80%, the population size 80 and number of generation 10.

6.2 Evaluation Measures

To evaluate the goodness of partition we used well known modularity Q (described in F4(P), section 4). Modularity is defined as the fraction of edges inside community decreased by the fraction of edges if the edges are randomly divided. Higher value of modularity means better partition. Modularity can have values from 0 to 1.

6.3 Real-world Networks

We used four well known data sets: Zackary karate club, Dolphin network, American college football and Kreb's political books.

Zackary karate club [27] is friendship network created by Zackary. Zackary observed 34 members of karate club for two years. Because of disagreements of trainer and administrator the club members divided into two groups of nearly equal size as shown in Fig 2.

The dolphin network consists of 64 dolphins living in Doubtful Sound, New Zealand and was created by Lussou [28]. Each two dolphins that have statistical frequent associations are represented by connected nodes. The dolphin network splits into two groups as shown in Fig.3.

American college football [30] contains the network of American football games between Division 1A colleges during regular season Fall 2000, as compiled by M. Girvan and M. Newman. The nodes in the network are the college football teams. There is a link between two teams if they played a game. Each team belongs to one conference. Conferences represent the real community structure. The teams tend to play more games with teams that are in the same conference. 11 communities shown in Fig 4 correspond to conferences.

The Krebs network [29] of political books consists of 105 political books from Amazon.com. Edges connect each two books that are often bought both by the same buyer. Books are divided usually to two big groups representing liberal or conservative political opinions (see Fig 5) and one small neutral group, with tendency to liberal political option or conservative.

6.4 Evaluation Results

Using the third optimization function increases the number of solutions for all tested third objective criteria (see Fig. 6-8). Fig. 6 shows function values (F1 and F2) and community numbers for all resulting partitions for Dolphin network using only two optimization criteria. Using optimization function F6 usually increases the number of communities much more than using modularity (F4) as optimization functions (see Figs. 7) and Table 2. Adding measure F6 that maximizes number of communities increase the number of communities from 10 in Fig. 6 to 17 in the resulted partition on Dolphin network in Fig.7. Using optimization function F5 gives also a lot of resulted partitions with large number of communities (Table 2).

![Fig.2. The Zackary karate club is usually split into two communities. The first number is community label and the second is a node label. All nodes with the same color and first number belong to the same community.](image)
Fig. 3. The Dolphin network is usually split into two communities. The first number is community label and the second is a node label. All nodes with the same color and first number belong to the same community.

Fig. 4. The American college football is usually split into 11 communities that correspond to conferences. All nodes with the same color and first number belong to the same community.

Fig. 5. The Krebs network of political books is divided into conservative, liberal and network community. The first number is community label and the second is a node label. All nodes with the same color and first number belong to the same community.

Fig. 6: Solutions (values of optimization criteria and numbers of communities) for Dolphin network using only two function (F1 and F2).

Fig. 7: Solutions (and values of optimization criteria) for Dolphin network using function F6 as the third optimization function. Values of all used functions and the obtained number of communities are shown.

Fig. 8: Solutions for Zackary karate club network using F4 function (1-modularity) as the third optimization criteria.
Table I: Best modularity results by our method and Girvan and Newman’s algorithm GN for real data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Modularity</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Avg Max</td>
<td>GN</td>
<td></td>
</tr>
<tr>
<td>Zackary karate club</td>
<td>0.415</td>
<td>0.393</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>Dolphins network</td>
<td>0.505</td>
<td>0.461</td>
<td>0.495</td>
<td></td>
</tr>
<tr>
<td>American Coll. Football</td>
<td>0.515</td>
<td>0.423</td>
<td>0.577</td>
<td></td>
</tr>
<tr>
<td>Kreb’s books</td>
<td>0.5</td>
<td>0.437</td>
<td>0.502</td>
<td></td>
</tr>
</tbody>
</table>

Table II: Using different functions as the third optimization function for Dolphin data set. Because the measure values fluctuate, we smooth these data with the mid-value method to illustrate them more clearly.

<table>
<thead>
<tr>
<th>The third optimization function</th>
<th>Number of communities</th>
<th>Number of partitions with number of communities&gt;6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>F4</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>F5</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>F6</td>
<td>14</td>
<td>2</td>
</tr>
</tbody>
</table>

F3 (Flake-ODF) considers the degree of nodes in a community. Experimental results do not include this optimization function, since it is very prone to divide the whole network as one community, when its objective value reach the smallest value (i.e., approximate to 0).

For each upper real-world network, we used optimization functions F1 and F2 and the different third optimization function. For each tested network the algorithm was executed 10 times and the modularity has been computed. The optimization function F4 gives the best modularities. The best values and average best values of modularity have been selected and shown in Table 1. The proposed multi-objective genetic algorithm successfully detects the network structure and it is competitive with GN. In fact, on the Zackary’s Karate Club the proposed method found the exact solution for all the 10 runs with average modularity value of 0.393, while the GN method obtained a modularity of 0.380. Also on the Dolphins network the proposed method found the exact solution in all the 10 runs with maximal modularity value of 0.505, and average max modularity 0.461, while the GN method obtained a modularity of 0.495. On the the Krebs’ network we obtained maximal best modularity nearly equal to GN.

7.Conclusion

We speed the community detection process by initializing one third of individuals with starting values that are close to possible solutions and with the use of three objective functions by the genetic algorithm. This enables wide spectrum of solutions, which does not require a priori knowledge of the number of communities, and no other threshold value. We have tested our algorithm’s accuracy on known real-world network datasets.

Acknowledgment

This work was supported by the Slovenian Research Agency (grant numbers: P2-0041, J2-5479).

References


