Genetic Algorithms for Multi-Objective Community Detection in Complex Networks

Ahmed Ibrahem Hafez1, 3, Eiman Tamah Al-Shammari2, Aboul ella Hassanien3, 4, Aly A. Fahmy4

1CS Dept., Faculty of Computer and Information, Minia University, Egypt

[ah.hafez@gmail.com](mailto:ah.hafez@gmail.com)

2Faculty of Computing Science and Engineering, Kuwait University

3Scientific Research Group in Egypt (SRGE), Egypt

4Faculty of Computers and Information, Cairo University, Egypt

[aboitcairo@gmail.com](mailto:aboitcairo@gmail.com), [aly.fahmy@gmail.com](mailto:aly.fahmy@gmail.com)

**Abstract.** Community detection in complex networks has attracted a lot of attention in recent years. Communities play special roles in the structure-function relationship. Therefore, detecting communities (or modules) can be a way to identify substructures that could correspond to important functions. Community detection can be viewed as an optimization problem in which an objective function that captures the intuition of a community as a group of nodes with better internal connectivity than external connectivity is chosen to be optimized. Many single-objective optimization techniques have been used to solve the detection problem. However, those approaches have drawbacks because they attempt to optimize only one objective function, this results in a solution with a particular community structure property. More recently, researchers have viewed the community detection problem as a multi-objective optimization problem, and many approaches have been proposed. Genetic Algorithms (GA) have been used as an effective optimization technique to solve both single- and multi-objective community detection problems. However, the most appropriate objective functions to be used with each other are still under debate since many similar objective functions have been proposed over the years. We show how those objectives correlate, investigate their performance when they are used in both the single- and multi-objective GA, and determine the community structure properties they tend to produce.

**Keywords:** Social networks, Community detection, Genetic Algorithms, communities’ quality measures

1. Introduction

Networks are used to model and represent many real world systems. This fact has made complex network analysis a popular research area. Collaboration networks, the Internet, biological networks, communication networks, and social networks are just some examples of such complex networks. A common feature of complex networks is community structure [[1](#Gir)], i.e. groups of nodes in the network that are more densely connected internally than with the rest of the network. Communities play special roles in the structure-function relationship, and detecting communities (or modules) can be a way to identify substructures that may correspond to important functions in the network. Therefore, uncovering or detecting community structure is one of the most important problems in the field of complex network analysis.

Many methods have been developed for the community detection problem. These methods use tools and techniques from disciplines like physics, biology, applied mathematics, computer and social sciences. Results of a recent survey can be seen in [[2](#SFo)].

One of the most popular algorithms proposed so far is the Girvan-Newman algorithm, which introduces a divisive method that iteratively removes the edge with the greatest betweenness value [[3](#New)]. Some improved algorithms have also been proposed [[4](#Rad),[5](#Cla1)]. These algorithms are based on a foundational measure criterion of community, Modularity which measures the number of within-community edges relative to a null model, which is a popular quality function that was proposed by Newman [[3](#New)]. The larger the Modularity value, the more accurate the community partition. Consequently, community detection becomes a Modularity optimization problem. Because the search for the optimal (largest) Modularity value is an NP-complete problem, many heuristic search algorithms, such as extremal optimization, simulated annealing, and genetic algorithms (GA), have been applied to solve the optimization problem [[2](#SFo)].

The remainder of this chapter is organized as follows. In Section 2 we define the community problem as a single-objective and multi-objective optimization problem and introduce the objective functions used in our GA. In Section 3 we discuss related work. We describe our GA, its genetic representation, and the operations used by the algorithm in Section 4. Section 5 reviews our experimental results for the single-objective and multi-objective cases on real life social networks and a synthetic network and apply the GA on a case study from Facebook social network. We then offer conclusions and suggestions for future work.

1. The community detection problem

A social network SN can be modeled as a graph ***G*** = (***V***, ***E***) where ***V*** is a set of vertices, and ***E*** is a set of edges that connect two elements of ***V***. A community structure ***S*** in a network is a set of groups of vertices having a high density of edges among the vertices and a lower density of edges between different groups. The problem of detecting *k* communities in a network, where the number *k* is unknown, given a quality measure of communities **F**(***S***), can be formulated as finding a partitioning of the nodes in *k* subsets that best satisfy the quality measure **F**(***S***). The problem can be viewed as an optimization problem in which one usually wants to optimize the given quality measure **F**(***S***). We will describe the most popular quality measures that have been used to detect communities in the following subsection.

* 1. Single objective Community detection

We can view the community detection problem as a single objective optimization problem.

A single objective optimization problem (**Ω**;**F**) is defined as

min **F**(***S***), s.t ***S*** ϵ **Ω** ()

where **F**(***S***) is an objective function that needs to be optimized and **Ω** = {***S***1, ***S***2, …, ***S****k*} is the set of feasible community structures in a network. We assume that all quality measures need to be minimized without loss of generality. So we want to find community structure ***S*** that maximizes or minimizes the quality measure **F**(***S***) according to the definition of the quality measure. A formal definition of the optimization problem is given in [[6](#Shi1)].

* 1. Multi-objective Community detection

Recently, researchers have treated the community detection problem as a multi-objective optimization problem where, given a set of quality measures **F**1(***S***), **F**2(***S***)…, **F**t(***S***), we want to find community structure ***S*** that simultaneously optimizes each quality measure.

A multi-objective optimization problem (**Ω**; **F**1, **F**2, … , **F**t) is defined as

min **F***i*(***S***), *i* = 1,2, … , t; s.t ***S*** ϵ **Ω** ()

Since the goal is to optimize a set of competing objectives optimized simultaneously, there is not one unique solution to the problem. A set of solutions is found through the use of the Pareto optimality theory [[7](#MEh)]. Given two solutions ***S***1 and ***S***2 ϵ **Ω** , solution ***S***1 is said to dominate solution ***S***2, denoted as ***S***1 ≺ ***S***2, if and only if

∀*i*: **F***i*(***S***1) ≤ **F***i*(***S***2) and ∃*i* s.t. **F***i*(***S***1) < **F***i*(***S***2) ()

Multi-objective optimization aims to generate and select non-dominated solutions, which are called Pareto-optimal solutions. It is worth noting that Pareto-optimal solutions, as outlined in [[8](#Jul)], usually include the optimal solutions obtained by single-objective GA when applied to the clustering problems. This will be explained in more detail in Section 5.

Genetic algorithm first proposed in [[9](#Hol)] is an optimization method applied to artificial intelligence problems that mimics the process of natural evolution. Genetic algorithms belong to a larger class of evolutionary algorithms, which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. It is a practical method, particularly when the solution space of a problem is very large and an exhaustive search for the exact solution is impractical. In GAs, potential candidate solutions in the solution set should be represented in a suitable data representation. Each candidate in the solution set, which is called a chromosome, represents a possible solution to the problem. The algorithm tries to find the candidate solution with the best fit. To improve the quality of the candidate solutions, the algorithm uses genetic operations, such as point mutation (random variations of some parts of the chromosome) and crossing over (generating new chromosomes by merging parts of existing chromosomes), on possible candidate solutions for a predefined number of iterations. At the outset, the algorithm randomly initializes the chromosomes. Then, for a number of iterations, it uses an objective function to assign a fitness value to evaluate each candidate solution’s relative ability to solve the problem. The GA then reproduces candidate solutions for a new population that will be used in the next iteration by performing crossover between candidates selected according to their fitness values. It also applies some random mutation to candidates. GAs are fast algorithms for converging a problem to a smaller solution space, and if the algorithm has a good objective function, it produces near optimal solutions. The power of a GA is the cross-over mechanism, which produces better next generation of candidate solutions.

The above description of a GA only employs one objective function to describe how good a solution is. Hence, only one objective can be optimized. However, most real-world problems involve simultaneous optimization of several and often competing objectives. For example, in the community detection problem we want to find communities that contain a high density of internal connections inside each community and have a low number of external connections between nodes from different communities, i.e., low Cut Ratio. GAs have also been applied to multi-objective optimization problems, and many algorithms have been proposed to deal with the problem using the Pareto optimality theory [[7](#MEh)]. When there are many, possibly conflicting, objectives to be optimized simultaneously, there is no longer a single optimal solution but rather a whole set of possible solutions of equivalent quality, and as we state before Multi-objective optimization aims to generate and select non-dominated Pareto-optimal solutions.

Due to the ability of multi-objective optimization to find multiple Pareto-optimal solutions in a single run, a number of multi-objective GAs have been suggested [[10](#NSr),[11](#27D),[12](#Zit),[13](#Zit1),[14](#Dav)] over the past decade. The Non-dominated Sorting Genetic Algorithm (NSGA) proposed by Srinivas and Deb [[15](#Kal)] was one of the first multi-objective GA algorithms. An improved and faster algorithm was proposed in [[11](#27D)]; the Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, which we will use in this work, overcomes some of the problems associated with the NSGA. Other algorithms in [[12](#Zit),[13](#Zit1)] use a Strength Pareto Approach, such as the Strength Pareto Evolutionary Algorithm (SPEA) proposed in [[12](#Zit)] and the Pareto Envelope-based Selection Algorithm (PESA) proposed in [[14](#Dav)].

* 1. Objectives

In a GA, the objective function plays an important role in the evolution process. It is the “steering wheel” in the process that leads to good candidate solutions. For the community detection problem, many objective functions have been proposed to capture the intuition of communities, and there is no straightforward way to compare these objective functions based on their definitions.

Here we state objective functions that capture this intuition and/or are popular in the literature, and can potentially be used for community detection. A detailed description of objective functions can be found in [[16](#Les10)] and a similarity comparison can be found in [[16](#Les10),[17](#Chu)].

In the following quality measures, the lower the value of **F**(***S***) the better the community structure:

* + Conductance [[16](#Les10)] measures the fraction of total edge volume that points outside the cluster.
  + Expansion [[16](#Les10)] measures the number of edges per node that point outside the cluster.
  + Internal Density [[16](#Les10)] is the internal edge density of the cluster.
  + Cut Ratio [[16](#Les10)] is the fraction of all possible edges leaving the cluster.
  + Normalized Cut [[18](#Shi97),[16](#Les10)] is the normalized fraction of edges leaving the cluster.
  + Maximum-Out Degree Fraction (ODF) [[19](#Fla00),[16](#Les10)] is the maximum fraction of edges of a node pointing outside the cluster.
  + Average-ODF [[19](#Fla00),[16](#Les10)] is the average fraction of node edges pointing outside the cluster.
  + Flake-ODF [[19](#Fla00),[16](#Les10)] is the fraction of nodes in ***S*** that have fewer edges pointing inside than outside of the cluster.

In the following quality measures, the higher the value of **F**(***S***), the better the community structure:

* + Modularity [[3](#New)] measures the number of within-community edges relative to a null model of a random graph with the same degree distribution.
  + Community Score [[20](#Cla)] measures the density of sub-matrices based on volume and row/column means.
  + Community Fitness [[21](#ndr)] is the ratio between the total internal degrees of the nodes belonging to that community and the sum of the total internal and external degrees of the nodes belonging to that community.
  + Surprise [[22](#Ald)] compares the number of links within and between communities in a partition with the expected number of links in a random network with the same distribution of nodes per community.

All but two of the quality measures are parameter free, i.e., calculation only depends on the network. Community Score and Community Fitness both have a positive real-valued parameter that controls the size of the communities.

1. Related work

GAs have been used as an effective optimization technique for community detection. [[23](#Tas),[24](#Shi)] used GAs to optimize the network modularity proposed by Girvan and Newman [[3](#New)]. Pizzuti proposed another GA to optimize the Community Score criterion [[25](#Piz),[20](#Cla)] and used the locus-based adjacency representation and uniform crossover for the genetic representation and the genetic operation. These algorithms have the advantage of automatically determining the number of communities during the evolutionary process. However, they also have a resolution limit, since a single objective is optimized.

A different approach is described in [[26](#Fir)]; a random walk distance measure between graphs is integrated in a GA to cluster social networks. They use the *k*-medoids as the genetic representation in which each community center is represented by one of the nodes of the network. This means that *k*, the number of communities, must be known in advance.

More recently, researchers have applied multi-objective optimization techniques to the community detection problem using multi-objective evolutionary algorithms [[6](#Shi1),[27](#Piz1),[28](#Roh),[29](#Haf12)]. Pizzuti [[27](#Piz1)] proposed a Multi-objective Genetic Algorithm (MOGA) for community detection in networks (MOGA-Net) based on NSGA-II [[11](#27D)] that simultaneously optimizes the Community Score and Community Fitness. Agrawal [[28](#Roh)] proposed a bi-objective community detection method also based on NSGA-II [[11](#27D)]. However, Agrawal used the Community Score and Modularity as the two objectives. Shi et al. [[6](#Shi1)] proposed a new multi-objective evolutionary algorithm based on the Pareto Envelope-based Selection Algorithm version 2 (PESA-II) [[30](#DCo)] and used the modularity objective to drive two new objectives and try to minimize the two objectives to find community structure.

1. Genetic algorithms for community detection

In this section we describe the GAs, genetic representation, and genetic operations used in this work. For both single- and multi-objectives we have adopted the genetic representation and genetic operations proposed in [[20](#Cla)]. We describe the various stages of the GA in the following subsections.

* 1. Genetic Representation

The algorithm uses the locus-based adjacency representation proposed in [[31](#YPa)] In this representation, each individual chromosome consists of *n* genes *g*1, *g*2, . . . ,*g*n and each gene can take values *j* in the range {1, . . .,*n*}. Where *n* = |***V***| is the number of nodes in the network, and a value *j* assigned to the *i*-th gene is interpreted as a link between the nodes *i* and *j*. This means that, in the detected community structure, *i* and *j* will be in the same community. A further decoding step is necessary to identify all communities. The advantages of this representation are that the number *k* of communities is automatically determined by the number of components contained in an individual and determined by the decoding step, and the decoding step can be achieved in linear time, as mentioned in [[32](#THC)].

* 1. Initialization

A random generation of individuals could generate components that are disconnected in the original graph. Pizzuti [[20](#Cla)] proposed the term safe initialization to describe a process where each gene *i* is assign to a value *j* from the *i*-th node’s neighbors. Here we use the same initialization steps used in [[20](#Cla)]. Then, with a probability of (1 – mutation Rate) of the population size, we select a ƞ percent of the genes, and for each selected gene i we assign it to itself, and for all its neighbors we assign the value *i*. Consequently, node *i* and its neighbors will be in the same community. After a series of trials, we found that a value of 0.5 for ƞ achieved a good result.

* 1. Uniform Crossover

Given two parents, a random binary vector is created. Uniform crossover then selects the genes where the vector is a 1 from the first parent and where the vector is a 0 from the second parent. These genes are then combined to form the new child.

* 1. Mutation

The mutation operator that randomly changes the value of a randomly chosen gene causes a useless exploration of the search space. Therefore, as in the initialization step, we randomly select a ɱ percent of the genes and for each gene *i* we randomly change its value to *j* such that node *i* and *j* are neighbors. After a series of trials, we found that a value of 0.1 for ɱ achieved a good result.

* 1. Objectives

As mentioned in Section 2, many community quality measures have been proposed over the last several years, and some are similar in behavior. We apply the algorithm to each quality measure as the optimization objective and compare the results to gain more insight into the definition of each objective and its properties.

1. Experimental Results

In our experimental setup, we employed a standard single-objective GA algorithm with a roulette selection function and elitism. The algorithm was implemented in a .NET environment using C# in the single-objective GA. In the multi-objective case we used NSGA-II [[11](#27D)] implemented in the MATLAB Genetic Algorithm and Direct Search Toolbox as the multi-objective GA. For more description reader may see [[15](#Kal)].

We tested the algorithm on a synthetic data set and two real social networks and finally we show the effect of the algorithm in detecting communities in online social network. To compare the accuracy of the resulting community structures, we used Normalized Mutual Information (NMI) to measure the similarity between the true community structures and the detected ones. NMI is a similarity measure proved to be reliable by Danon et al. [[33](#LDa)]. The NMI similarity measure is inspired from information theory and is based on defining a confusion matrix **N**, where the rows correspond to the real communities **A** and the columns correspond to the found communities **B**. The members of **N**, *N*ij, are simply the number of nodes in the real community *i* that appear in the founded community *j*. The number of real communities is denoted as *c*A and the number of founded communities is as denoted *c*B .The sum over row *i* of matrix *N*ij is denoted as *N*.i , and the sum over column *j* is denoted as *N*j.. Based on information theory, a measure of similarity between the partitions is then:

()

We calculated the average NMI for many runs of the algorithm for each objective and compared the results.

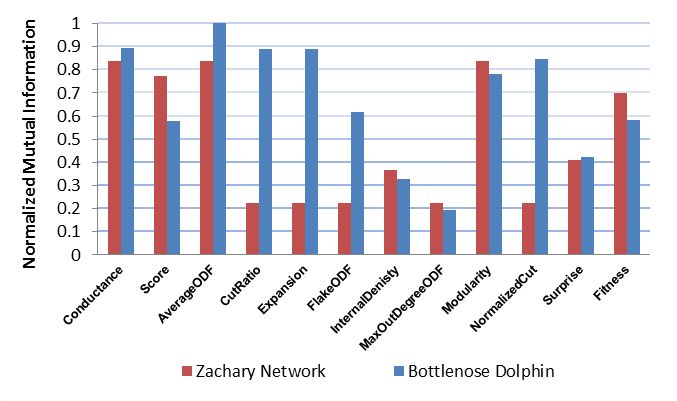
**Real Social Network**: We tested the algorithm for each objective on two real life data sets, the Zachary’s Karate Club and the Bottlenose Dolphins. Both data sets have been well studied in the literature (see [[34](#Net12)]). The Zachary Karate Club data, which was first analyzed in [[35](#Zac)], contains the community structure in a karate club. The network consists of 34 vertices and 78 edges. The network is divided into two approximately equal groups. The Bottlenose Dolphin data was compiled by Lusseau [[36](#Lus)] and is based on observations over a period of seven years of the behavior of 62 bottlenose dolphins living in Doubtful Sound, New Zealand. A relationship between two dolphins was established by their statistically significant frequent association. The network split naturally into two large groups.

**Synthetic network**: We used the benchmark proposed by Girvan and Newman in [[1](#Gir)]. The network consists of 128 nodes divided into four equal-size communities. Edges are placed between vertex pairs at random such that zin + zout = 16, where zin and zout are the internal and external degrees of a node with respect to its community respectively. If zin > zout, the neighbors of a node inside its group are greater than the neighbors belonging to the other three groups then the network has a strong community structure. Thus, a good algorithm should discover the communities up to zout = 8.

* 1. Single objective community detection

We began by applying the single objective GA to the community detection problem. We ran each objective separately in the GA.

The results for the real life social networks, Zachary’s Karate Club and the Bottlenose Dolphins, are presented in Fig. 1. The graph shows the average NMI over 30 runs of the GA for each objective. We employed standard parameters for the GA: a crossover rate of 0.9, a mutation rate of 0.4, the elite reproduction was 10% of the population size, the population size was 60, and the number of iterations was 35. Three objectives, Conductance, Average-ODF, and Modularity, achieved a NMI value above 0.8 for both social networks. Two objectives, Community Score and Community Fitness, also achieved a good NMI value above 0.5 for both networks. For the synthetic network, we generated 15 networks for each value of zout in a range from 0 to 6 and ran the algorithm 10 times. We calculated the average NMI value for the 10 runs and then calculate the average NMI value for each value of zout for the 15 networks. Figure 2 shows the NMI values for the synthetic network for each objective separately. We used the same values for the standard parameters for the GA, except for the number of iterations (generation). Because the synthetic network is larger than the social networks, we set the number of iterations to 100.



**Fig. .** NMI values obtained by each objective on Zachary Karate Club Network and Bottlenose Dolphin Network

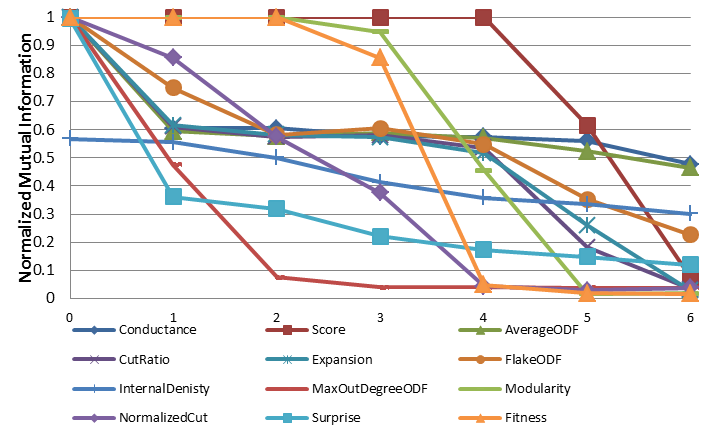
We found that the Community Score objective achieved a good result until zout = 5. The Community Fitness and Modularity objectives achieved a good result until zout = 3. The Internal Density objective failed to detect the correct community structure for the network when zout = 0; at this value the network consisted of four disconnected communities. We found that all objectives, except Conductance, Average-ODF, and Internal Density, decreased to a NMI value less than 0.1 when zout reaches 6.

### Objectives Stabilities

GA is a non-deterministic algorithm, due to the random nature of the GA i.e., it may produce a different solution each run. The objective function is responsible for directing the algorithm to global/local maxima or global/local minima based on the definition of the objective over the solution space, usually we want the objective to direct the algorithm to a global maxima but it may fail sometimes to do that. A good objective function should assign the same fitness score to two solutions if they are the same or very similar. If two completely different solutions have the same fitness score obtained by the objective function, the algorithm will tend to produce a different solution in each run.

We define a stable objective as an objective that tends to produce similar community structures over many runs of the GA. Figure 3 shows the average similarity of solutions obtained by an objective in term of the NMI similarity measure. A low value indicates that, over many runs, the solutions vary considerably; a high value indicates that the solutions are much more similar to each other.

We ran the algorithm many times on each network, Zachary’s Karate Club, the Bottlenose Dolphins, and the synthetic network, for a value of zout = 3 and calculated the average NMI for each pair of solutions for each network. We observed that Community Score, Community Fitness, and Modularity achieved high values for all networks. This indicates that they were more stable than the rest of objectives because they produced similar results for different runs. Most of the objectives achieved high values for the two real life social networks, as shown in Fig. 3.



**Fig. .** NMI values for each objective on the GN Benchemark for zout range from 0 to 6

### Objectives Similarities

In Section 2, we showed and defined many objectives, and stated that they are somehow similar in definition. Additionally, when deployed with the GA, some tend to produce equal or similar results.

**Fig. .** Objectives Stabilities in term of NMI obtained by running GA over real life data sets; Zachary Karate Club Network and Bottlenose Dolphin Network, and GN benchmark for zout = 3 .

To understand these similarities among objectives, we compared the solutions obtained by each objective with the solutions obtained by the remainder of the objectives. The average NMI values are shown in Fig. 4. A low value (dark cells) indicates low similarity, and a high value (white cells) indicates high similarity.

For the Zachary Karate Club Network, Community Fitness produced a result similar to Modularity, Conductance, and Community Score. Additionally, Expansion, Cut Ratio, Max-ODF, Flake-ODF, and Normalized Cut also show similar results. Conductance and Modularity produce the exact solution. For the Bottlenose Dolphin network, we found there is high similarity between Conductance, Average-ODF, Cut Ratio, Expansion, Normalized Cut, and Modularity. For the GN Network, there is a lot of variation in the output of all objectives except Community Score and Modularity, which produced similar results and good community structure.

### Network Modularity and other objectives

We also compared objectives based on network Modularity because it is a popular community quality measure used extensively in community detection.

First, it should be noted that the Modularity of the optimal community structure of Zackary is 0.3 and is 0.33 for Dolphin. For the GN benchmark, we show the average Modularity of the optimal community structure of the generated network for different zout values in Table. 1. We observed that when the zout valueincreased, the Modularity of the original community structure decreased. When the zout value increased, the number of external connections for each node increased, which led to increased randomness in the network. Therefore, the original community structure was no longer a strong structure, and there was a high probability that other community configurations with high Modularity value existed.

|  |  |
| --- | --- |
| C:\Users\User\Pictures\4a.png | C:\Users\User\Pictures\4b.png |
| C:\Users\User\Pictures\4c.png | Community similarities obtained by different objectives:   1. Zachary Karate Club Network 2. Bottlenose Dolphin Network 3. GN-Benchmark for zout = 3   Diagonal entries correspond to objective stabilities |

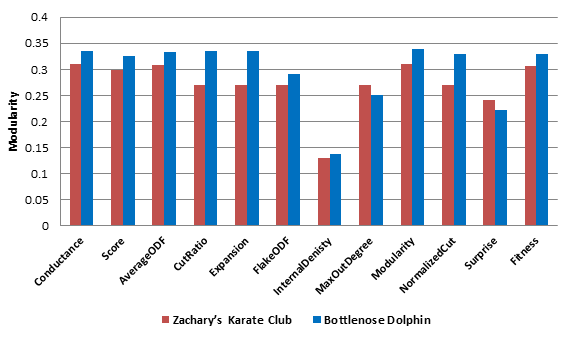
**Fig. .** Community similarities obtained by different objectives

**Table .** Modularity value for optimal community structure of the GN Network for different zout values

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Zout | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Modularity | 0.4374 | 0.3975 | 0.3662 | 0.3391 | 0.3036 | 0.2696 | 0.238 | 0.2104 |

The same experiment was conducted as before, and for each run we calculated the Modularity of the community structure returned by each objective. We then calculated the average Modularity for all runs. In Fig. 5, we show the Modularity of the community structure returned by different objectives for the real social network. From the results, it can be seen that most objectives returned community structures with good Modularity values.

For the Dolphin network, only the Average-ODF objective detected the optimal community structure of the network with a Modularity value of 0.333. Additionally, the Modularity objective found a community structure with a Modularity value of 0.338, which is better than the optimal community structure of the network. Both Cut Ratio and Expansion found community structures with a Modularity value of 0.335, which is also better than the optimal solution in terms of Modularity. In Fig. 6, we visualize[[1]](#footnote-1) the solution returned by Average-ODF, which is also the optimal solution of the network, and solutions returned by the Modularity, Cut Ratio, Expansion, Normalized Cut, and Surprise objectives.



**Fig. .** Modularity values achieved by each objective for Zachary Karate Club network and Bottlenose Dolphin network

* 1. Multi-objective community detection

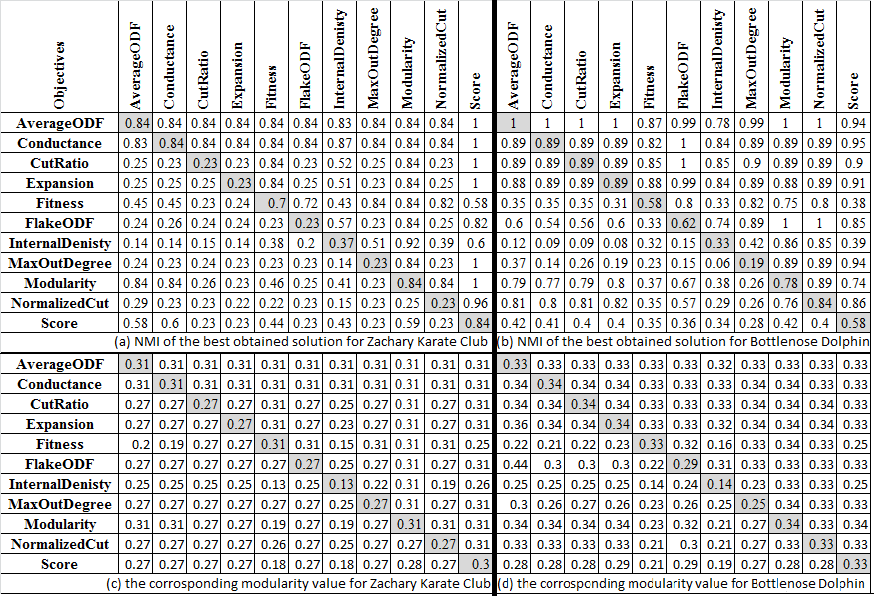
We applied the multi-objective GA to the community detection problem. We restricted the number of objectives used in the algorithm to two and studied the algorithm’s performance for each pair of objectives. We excluded the Surprise objective because it is computationally expensive as it requires many binomial coefficient calculations and does not scale well for large networks. Additionally, the Surprise objective was excluded because it did not achieve good results in the single objective scenario.

Figure 7 shows the average NMI over 15 runs of the GA for each pair of objectives and the corresponding average Modularity value of each solution for the Zachary’s Karate Club and the Bottlenose Dolphins social networks. We employed standard parameters for the GA; a crossover rate of 0.8, a mutation rate of 0.2, the elite reproduction rate was 10% of the population size. We also employed a binary tournament selection function. The population size was 100, and the number of generations was 30.

|  |  |
| --- | --- |
| C:\Users\User\Pictures\6a.jpg  a) Average-ODF | C:\Users\User\Pictures\6b.jpg  b) Modularity |
| C:\Users\User\Pictures\6c.jpg  c) Surprise | C:\Users\User\Pictures\6d.jpg  d) Normalized Cut |
| C:\Users\User\Pictures\6e.jpg  e) CutRatio / Expansion | 1. Average-ODF : NMI = 1 , Modularity = 0.333 2. Modularity : NMI = 0.82 , Modularity = 0.338 3. Surpirse : NMI = 0.45 , Modularity = 0.22 4. Normalized Cut : NMI = 0.9, Modularity = 0.329 5. CutRatio and Expansion : NMI = 0.88, Modularity = 0.335 |

**Fig. .** Community strutures obtianed by appling Single-objective GA for Bottlenose Dolphin Network

For each run, the Pareto-optimal set returned by the algorithm was investigated. The best community structure was selected and compared to the optimal known solution of the network in terms of the NMI similarity measure. Additionally, the worst community structure in the set was also compared to the optimal solution of the network. Modularity values for both best and worst structures were also calculated.



**Fig. .** NMI values and Modularity values for the best and the worst community structure in the Pareto-optimal set for (a) and (c) the Zachary Karate Club Network, (b) and (d) the Bottlenose Dolphin Network. The upper triangular area of the matrix shows the values for the best solution for each pair. The lower triangular area shows the values for the worst solution for each pair. The diagonal entries correspond to the single objective case.

Figures 7.a and 7.b show the NMI values for the best and worst community structures in the Pareto-optimal set when we applied the multi-objective GA using two objectives for the Zachary Karate Club Network and the Bottlenose Dolphin Network, respectively. The NMI values for the best solution are in the upper triangular area of the matrix. The NMI values for the worst solution compared to the optimal solution are in the lower triangular area of the matrix. The diagonal entries correspond to the single objective case. Figures 7.c and 7.d show the corresponding Modularity values of the best and worst solutions in terms of the NMI for the Zachary Karate Club Network and the Bottlenose Dolphin Network respectively. Modularity values for the best solutions are in the upper triangular area of the matrix, and Modularity values for the worst solutions compared to the optimal solution in term of NMI are in the lower triangular area. The diagonal entries correspond to the Modularity values for the single-objective case.

Figure 8 visualizes some of the results of the multi-objective case for the Bottlenose Dolphin Network. Figure 8.a shows a result determined by the Modularity and Normalized Cut objectives with a Modularity value of 0.338, and Fig. 8.b shows the result found by the Score and Fitness objectives with a Modularity value of 0.329. Figure 8.c shows the result found by the Score and Modularity objectives with a Modularity value of 0.336.

|  |  |
| --- | --- |
| C:\Users\User\Pictures\8a.png  a) Modularity and Normalized Cut | C:\Users\User\Pictures\8b.png  b) Score and Fitness |
| C:\Users\User\Pictures\8c.png  c) Score and Modularity | Community struture for Bottlenose  Dolphin Network found by:   1. Modularity and Normalized Cut : NMI = 0.6975, Modularity = 0.3384 2. Score and Fitness : NMI = 0.8387, Modularity = 0.3332 3. Score and Modularity : NMI = 0.6861, Modularity = 0.3358 |

**Fig. .** Community strutures obtianed by appling Multi-objective GA for Bottlenose Dolphin Network

When used with other objectives, Community Score achieved good NMI results for both networks, except when used with Community Fitness and Internal Density objectives. In addition, Community Fitness achieved good NMI results for both networks, except when used with Community Score and Internal Density objectives. Average-ODF and Conductance achieved very promising NMI results for both networks when used with any objective.

The NMI results for the best and worst solution for the synthetic network are shown in Fig. 9, and in Fig. 10, we show the corresponding Modularity values for these solutions. Here, we employed the same standard parameter values used for the GA, except for the number of iterations, which was set to 100. We applied each pair of objectives to the network for zout values ranging from 0 to 7. As was done previously, we generated different networks for each zout value and took the average result from each run. We only show the results for zout values from 1 to 7.

For zout = 0, all pairs were able to detect the correct network community structure, except for Internal Density with Community Fitness, which achieved NMI values of 0.6. For zout values from 1 to 3, when Community Score was applied with all objectives, the optimal community structure was detected, as it did for the single objective case. In addition, Community Fitness correctly detected the optimal community structure when applied with all objectives except for Internal Density. Modularity achieved good NMI results with values above 0.9 when applied with other objectives, except for Normalized Cut. For values of zout > 4, the performance of most objective pairs began to decrease; however, it was determined that Community Fitness achieved good results until zout = 6 when applied with other objectives. Community Fitness with Flake-ODF and Average-ODF achieved good results for zout = 7.

Figure 10.e shows some of the solutions having low NMI values (lower triangular area of the matrix) have a better Modularity value. This indicates that such solutions have good community structure. This is due to high zout values, which led to increased randomness in the network and the low Modularity value of the original community structure of the generated network (Table 1). This further indicates that the GN network benchmark is not sufficient for adequate testing.

One advantage of a multi-objective GA is that it returns a set of Pareto-optimal solutions. There is no best or worst solution; the most appropriate solution can be selected according to specific needs or other criteria. For example, the algorithm could run using Community Fitness and Flake-ODF, and the solution with the maximum network Modularity could be selected from the Pareto-optimal solutions set. Modularity has been selected as our reference when comparing determined community structures because it is a popular measure and most recent work in community detection is based on this quality measure.

* 1. Comparing with other algorithms

As mentioned before the Girvan-Newman (GN) algorithm [[3](#New)] and an improved GN algorithm proposed by Clauset, Newman, and Moore [[5](#Cla1)] are from the first community detection algorithms proposed to date. In this section we compare the results of those algorithms with the best result obtained by the GA in the single-objective case and the multi-objective case.

First we show the community structure found by the two algorithms for the Bottlenose Dolphin Network in Fig. 11. Results were obtained using NodeXL [[37](#Sta13)], a plugin for Microsoft Excel. However, the NMI calculations were done in a program that we wrote in C# to ease the comparison process.

|  |  |
| --- | --- |
| C:\Users\User\Pictures\9a.png | C:\Users\User\Pictures\9b.png |
| C:\Users\User\Pictures\9c.png | C:\Users\User\Pictures\9d.png |
| C:\Users\User\Pictures\9e.png | C:\Users\User\Pictures\9f.png |
| C:\Users\User\Pictures\9g.png |  |

**Fig. .** NMI values for the best and worst community structure in the Pareto-optimal set for GN benchmark for zout values from 1 to 7. The upper triangular of the matrix show the NMI values for the best solution for each pair. The lower triangular show the NMI values for the worst solution for each pair.

|  |  |
| --- | --- |
| C:\Users\User\Pictures\10a.png | C:\Users\User\Pictures\10b.png |
| C:\Users\User\Pictures\10c.png | C:\Users\User\Pictures\10d.png |
| C:\Users\User\Pictures\10e.png | C:\Users\User\Pictures\10f.png |
| C:\Users\User\Pictures\10g.png |  |

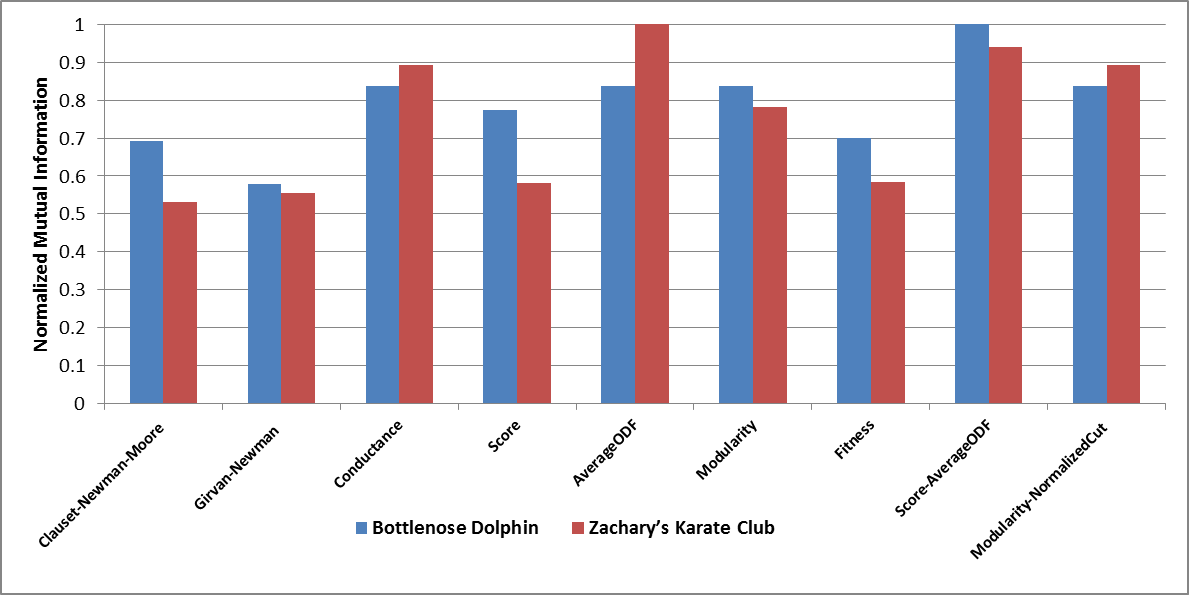
**Fig. .** Corresponding Modularity values for the best and worst community structures in the Pareto-optimal set from the GN benchmark for zout values from 1 to 7. The upper triangular areas of the matrices show the Modularity values for the best solution for each pair. The lower triangular areas show the Modularity values for the worst solution for each pair of objectives for the solution selected in Fig. 9.

|  |  |
| --- | --- |
| C:\Users\User\Pictures\11a.pnga) Clauset, Newman and Moore | C:\Users\User\Pictures\11b.png b) Girvan-Newman |

**Fig. .** Community struture founded by (a) Clauset, Newman and Moore and (b) Girvan-Newman algorithms for Bottlenose Dolphin Network

Figure 12 shows the NMI values of the solutions found by the two algorithms compared with some of our results using the GA for both real social networks.

In the single-objective case, we observe that the GA produced better results than the GN and Clauset algorithms for both networks for most of the objectives. In the multi-objective case, the multi-objective GA with the proper objective function also produced a very promising result for both networks.



**Fig. .** NMI comparision between best of GA results, Clauset-Newman-Moore algorithm, and Girvan-Newman algorithm

* 1. Case Study: Communities in online social network

Online social network such as Facebook or Google+ is a platform to build social networks or social relations among people who, for example, share interests, activities, backgrounds, or real-life connections. A social network service consists of a representation of each user (often a profile), his/her social links, and a variety of additional services. Social networking sites allow users to share ideas, pictures, posts, activities, events, and interests with people in their network due such online social behavior online communities are formed where online users tend to form communities that group users who share some comment interest. Users usually form groups or circles in online social network. Leskovec [[38](#McA12)] collects some data for the Facebook website -10 ego networks-. The data was collected from survey participants using a Facebook application [[39](#Les13)]. The dataset includes node features (profiles), circles, and ego networks. The ago network consist of a user’s –the ego node- friends and their connections to each other. Each ego network has a ground truth community structure that has been identified by the ego used himself by just tagging each of his friends. The problem with the ego network ground truth is that it was identified by the ego user himself, making it valid form the user point of view and most participants users miss classify some of his/her friends .

We combine the 10 Facebook ego networks from [[38](#McA12)] into one big network. We remove the ego nodes. The result network is undirected network which contain 3959 nodes and 84243 edges. There is no clear community structure for the network so we compare the result of the GA only in term of Modularity.

First we applied single-objective GA to the network for each objective. We employed standard parameters for the GA: a crossover rate of 0.9, a mutation rate of 0.4, the elite reproduction was 10% of the population size, the population size was 100, and the number of iterations was 400. Figure 13 shows the result Modularity value for the detected community structure of each objective and the number of communities per each community structure. Almost all Objectives produce a good community structure in term of Modularity for Facebook dataset. First Modularity and Community Fitness objectives produce a good community structure with a high Modularity value above 0.8. Then there are Conductance and Normalized Cut objectives produce community structures with Modularity value about 0.75. Only Internal density achieves a low modularity value as observed in Fig. 13.a. The number of communities’ *k* in each community structure produced by each objective is different. Figure 13.b shows the number of communities for the result of each objective. The community structures returned by Community Score and Internal density objective have more than 240 communities. The community structures returned by Modularity, Community Fitness and Normalized cut objectives have approximately the same size which in the range from 150 to 190. The rest of the objectives produce a community structure which has less than 90 communities.

|  |  |
| --- | --- |
| C:\Users\User\Pictures\13.png  (a) | C:\Users\User\Pictures\13b.png(b) |

**Fig. .** Modularity value for the detected community structure for each objective and the number of communities per each community structure

Despite that the number of communities seems somehow large, the distribution of nodes in each community is not uniform, we observe that a small subset of community tend to be large –contain most of the nodes- and other is very small which have less than 10 nodes. In Fig 14 we show the nodes distribution over communities for the result of two objectives Modularity and Conductance. We can observe that result community structure has about 5 large communities and 10 medium size communities and the rest of communities are very small which contain less than 15 nodes.

We visualize the best results of the single-objective GA for the Facebook dataset which are the result obtained by Modularity, Community Fitness and Conductance objectives. The community structure returned by Modularity is approximately the same as Community Fitness, the NMI comparison of both community structures obtained by Modularity and Community Fitness was 0.85. So we visualized the result for Modularity and Conductance only in Fig. 15. Visualization of this network has be done using Gephi [[40](#Bas)] via utilizing ForceAtlas2 layout algorithm [[41](#Mat)]. The network layout shown in Fig. 15 highlights the network community structure.

|  |  |
| --- | --- |
| C:\Users\User\Pictures\14a.png  (a) Modularity | C:\Users\User\Pictures\14b.png  (b) Conductance |

**Fig. .** Nodes distribution over communities for the result of two objectives Modularity and Conductance

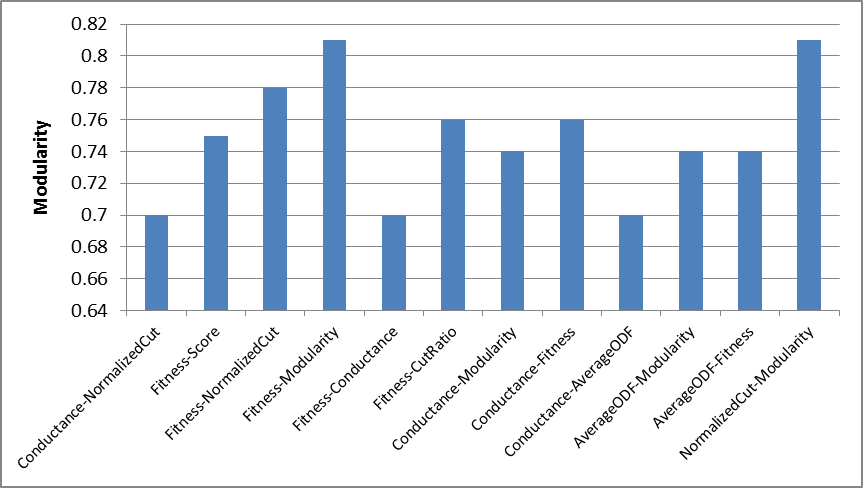
From Fig. 15 we can observe the existence of strong communities in the network in which almost all nodes are connected to each other. The community structure returned by Modularity objective is shown in Fig. 15.a; we can observe a clear community structure. The major difference between the two community structures shown in Fig. 15.a and Fig. 15.b is that Modularity objective was able to break about 4 large communities in Fig. 15.b into smaller communities which is better.

|  |  |
| --- | --- |
| D:\Master\Revised work\book chapter 1\1Mod.png  (a) Modularity | D:\Master\Revised work\book chapter 1\cond.png  (b) Conductance |

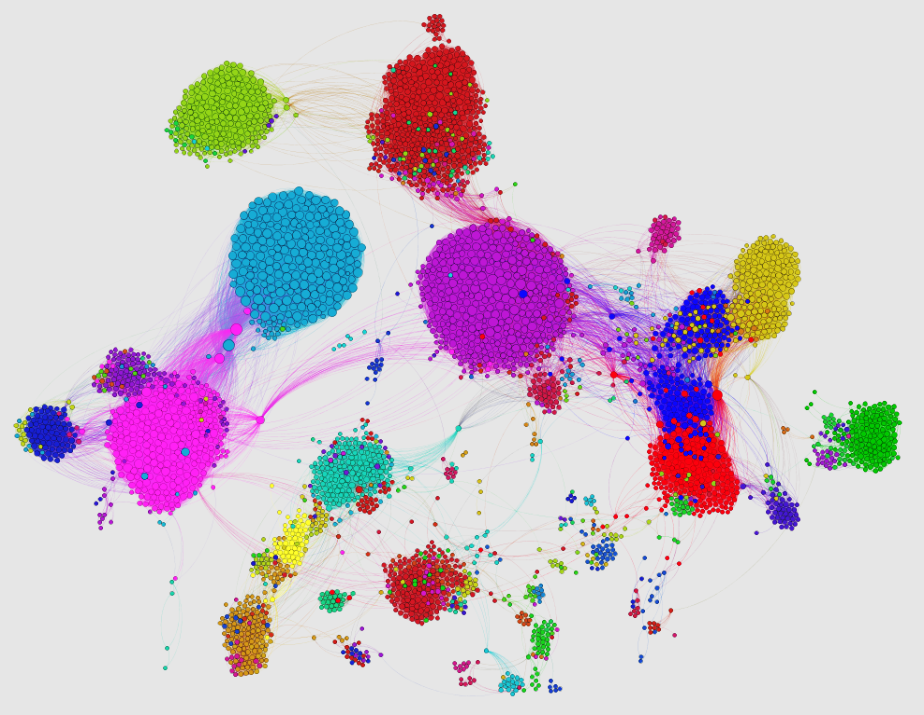
**Fig. .** Community structures of Facebook data sets as found by a) Modularity and b) Conductance objectives

Second we applied the Multi-Objective GA for each pair of objective on the Facebook dataset. We employed standard parameters for the GA; a crossover rate of 0.8, a mutation rate of 0.2, the elite reproduction rate was 10% of the population size, population size was 100, and the number of generations was 300. We show only the best result for a few pairs which prove to produce a good result when applied with each other. Fig. 16 summarizes the best result of Multi-Objective GA when applied to the Facebook dataset.

The result of Multi-objective GA when applied using two objectives inherits some properties of the result of each objective when used in single objective GA. For example when applied Conductance and Modularity the result community structure has 122 communities, however the community structure of Conductance only has 66 communities and the community structure of Modularity only has 180 communities. The fact that GA try to optimize the solution for both objective, lead to a result that average the result of both objective. We show the result community structure obtained by Conductance-Modularity in Fig. 17.



**Fig. .** Best result for the Facebook dataset obtained by applying Multi-Objective GA



**Fig. .** Community structure of Facebook dataset obtained by applying Multi-Objective GA using Conductance and Modularity objectives

1. Conclusion and future works

A GA as an optimization technique works effectively for the community detection problem in single-objective and multi-objective cases. However, performance is influenced directly by the objective function used in the optimization process. For the multi-objective case, we found that some objectives worked well together while others did not. However, it would be interesting to design new objectives, different from those used in this study, that are more suitable for the multi-objective genetic algorithm. Additionally, we could explore the performance of a different multi-objective optimization technique, such as multi-objective Bayesian optimization, for the community detection problem.

**References**

x

|  |
| --- |
| 1. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12), 7821–7826 (2002) |
| 1. Fortunato, S.: Community detection in graphs. Physics Reports 486(3–5), 75-174 (2010) |
| 1. Newman, M. E. J., Girvan, M.: Finding and evaluating community structure in networks. Physics Rev. E 69(2), 026113 (2004) |
| 1. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of USA 101(9), 2658-2663 (2004) |
| 1. Clauset, A., Newman, M., Moore, C.: Finding community structure in very large networks. Phys. Rev. E 70(6), 066111 (2004) |
| 1. Shi, C., Zhong, C., Yan, Z., Cai, Y., Wu, B.: A multi-objective optimization approach for community detection in complex network. In : IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain, pp.1-8 (2010) |
| 1. Ehrgott, M.: Multicriteria Optimization 2nd edn. Springer, Berlin (2005) |
| 1. Handl, J., Knowles, J.: An Evolutionary Approach to Multiobjective Clustering. IEEE Transactions on Evolutionary Computation 11(1), 56-76 (2007) |
| 1. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975) |
| 1. Srinivas, N., Deb., K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolitionary Computation 2(3), 221–248 (1994) |
| 1. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Parallel Problem Solving from Nature PPSN VI Lecture Notes in Computer Science 1917, 849-858 (2000 ) |
| 1. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3(4), 257-271 (Nov 1999) |
| 1. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2), 173-195 (2000) |
| 1. Corne, D., Knowles, J., Oates, M.: The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In : Proceedings of the Parallel Problem Solving from Nature VI Conference, pp.839--848 (2000) |
| 1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Ltd, England (2001) |
| 1. Leskovec, J., Lang, K., Mahoney, M.: Empirical Comparison of Algorithms for Network Community Detection. Proceeding of  19th international conference on World Wide Web (ACM WWW), 631-640  (2010) |
| 1. Shi, C., Zhong, C., Yan, Z., Cai, Y., Wu, B.: On selection of objective functions in multi-objective community detection. In : Proceedings of the 20th ACM international conference on Information and knowledge management, Glasgow, Scotland, UK, 2301-2304 (2011) |
| 1. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888--905 (1997) |
| 1. Flake, G., Lawrence, S., Giles, C.: Efficient Identification of Web Communities. In : Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 150--160 (2000) |
| 1. Pizzuti, C.: Ga-net: a genetic algorithm for community detection in social networks. In : Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X, Dortmund, Germany, 1081--1090 (2008) |
| 1. Lancichinetti, A., Fortunato, S., Kertesz, J.: Detecting the overlapping and hierarchical community structure of complex networks. arXiv:0805.4770v2 (2008) |
| 1. Aldecoa, R., Marín, I.: Deciphering Network Community Structure by Surprise. PLoS ONE 6, e24195 (September 2011) |
| 1. Tasgin, M., Bingol, H.: Community detection in complex networks using genetic algorithm. arXiv:cond-mat/0604419 (2006) |
| 1. Shi, C., Zhong, C., Yan, Z., Cai, Y., Wu, B.: A new genetic algorithm for community detection. Complex Sciences 5(2), 1298-1309 (2009) |
| 1. Pizzuti, C.: Community detection in social networks with genetic algorithms. In : Proceedings of the 10th annual conference on Genetic and evolutionary computation, Atlanta, GA, USA, pp.1137-1138 (2008) |
| 1. Firat, A., Chatterjee, S., Yilmaz, M.: Genetic clustering of social networks using random walks. Computational Statistics and Data Analysis 51(12), 6285–6294 (2007) |
| 1. Pizzuti, C.: A multi-objective genetic algorithm for community detection in networks. In : 21st International Conference on Tools with Artificial Intelligence, pp.379-386 (2009) |
| 1. Agrawal, R.: Bi-Objective Community Detection (BOCD) in Networks using Genetic Algorithm. Contemporary Computing, Communications in Computer and Information Science 168, 5-15 (2011) |
| 1. Hafez, A. I., Ghali, N. I., Hassanien, A. E., Fahmy, A. A.: Genetic Algorithms for community detection in social networks. In : 12th International Conference on, Intelligent Systems Design and Applications (ISDA), 2012 pp.460-465 (2012) |
| 1. Corne, D., Jerram, N., Knowles, J., Oates, M.: PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp.283–290 (2001) |
| 1. Park, Y., Song, M.: A genetic algorithm for clustering problem. In : Proceedings of the 3rd Annual Conference on Genetic Programming, pp.568–575 (1998) |
| 1. Cormen, T., Leiserson, C., Rivest, R., Stein, a.: Introduction to Algorithms. MIT Press (2001) |
| 1. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment 9, 09008 (2005) |
| 1. In: Network DataSets. (Accessed 2013) Available at: <http://www.personal.umich.edu/˜mejn/netdata> |
| 1. Zachary, W. W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33(4), 452-473 (1977) |
| 1. Lusseau, D.: The emergent properties of dolphin social network. Proceedings of the Royal Society of London. Series B: Biological Sciences 270(Suppl 2), S186-S188 (2003) |
| 1. In: Stanford Large Network Dataset Collection. (Accessed 2013) Available at: <http://snap.stanford.edu/data/index.html> |
| 1. McAuley, J., Leskovec, J.: Learning to Discover Social Circles in Ego Networks. In : NIPS, pp.548-556 (2012) |
| 1. Leskovec, J. In: Social Circles in Ego Networks. (Accessed 2013) Available at: <http://snap.stanford.edu/socialcircles/> |
| 1. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An Open Source Software for Exploring and Manipulating Networks. In : International AAAI Conference on Weblogs and Social Media, http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154, .2009 |
| 1. Jacomy, M., Heymann, S., Venturini, T., Bastian, M.: ForceAtlas2, A continuous graph layout algorithm for handy network visualization. Medialab center of research (2011) |

x

1. Community Visualization has been done using NodeXL plugin for Microsoft Excel from http://snap.stanford.edu/index.html. [↑](#footnote-ref-1)