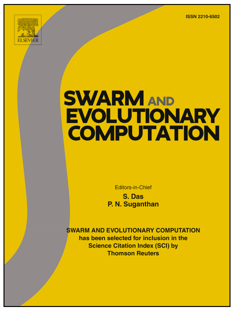


A multi-objective adaptive evolutionary algorithm to extract communities in networks

Qi Li, Zehong Cao, Weiping Ding, Qing Li

PII: S2210-6502(19)30417-1

DOI: https://doi.org/10.1016/j.swevo.2019.100629

Reference: SWEVO 100629

To appear in: *Swarm and Evolutionary Computation BASE DATA*

Received Date: 31 May 2019

Revised Date: 4 September 2019

Accepted Date: 27 November 2019

Please cite this article as: Q. Li, Z. Cao, W. Ding, Q. Li, A multi-objective adaptive evolutionary algorithm to extract communities in networks, *Swarm and Evolutionary Computation BASE DATA* (2020), doi: https://doi.org/10.1016/j.swevo.2019.100629.

A Multi-Objective Adaptive Evolutionary Algorithm to Extract Communities in Networks

Qi Li[a], Zehong Cao[b,d,*], Weiping Ding[c,d,*], Qing Li[e]

[a] *College of Mechanical and Electrical Engineering, Shaoxing University, Zhejiang, China*
[b] *Discipline of ICT, University of Tasmania, TAS, Australia*
[c] *School of Information Science and Technology, Nantong University, Jiangsu, China*
[d] *School of Computer Science, University of Technology Sydney, NSW, Australia*
[e] *College of Computer Science, Chongqing University, Chongqing, China*

Abstract

Community structure is one of the most important attributes of complex networks, which reveals the hidden rules and behavior characteristics of complex networks. Existing works need to pre-set weight parameters to control the different emphasis on the objective function, and cannot automatically identify the number of communities. In the process of optimization, there will be some challenges, such as premature and inefficiency. This paper presents a multi-objective adaptive fast evolutionary algorithm (F-SGCD) for community detection in complex networks. Firstly, it transforms the problem of community detection into a multi-objective optimization problem and constructs two objective functions of community score and community fitness. Secondly, an external elite gene pool is introduced to store non-inferior solutions with high fitness. At the same time, an adaptive genetic operator is executed to return a set of non-dominant solutions compromised between the two objective functions. Finally, a Pareto optimal solution with the highest modularity is selected and decoded to generate a set of independent subnetworks. Experiments show that the multi-objective adaptive fast evolutionary algorithm greatly improves the accuracy of community detection in complex networks, and can discover the

[*] Corresponding author
Email addresses: zehong.cao@uts.edu.au (Zehong Cao), dwp9988@163.com (Weiping Ding)

hierarchical structure of complex networks better.

Keywords: community detection, genetic algorithm, multi-objective, complex networks, adaptive

1. Introduction

Many complex systems in the real world, such as social networks, protein networks, and transmission networks, can be abstracted into complex networks, which have small world and scale-free characteristics [1]. At the same time, it has a characteristic of community structure, which is characterized by relatively close links within communities and relatively sparse links among different communities [2]. Through the discovery of the network community structure, we can get the internal structure of the network and the interaction relationship, which not only provides an effective way to solve practical problems but also dramatically reduces the complexity of the research. Therefore, it is of considerable significance to carry out community detection research.

The definition of community detection pursues two different objectives: maximizing the internal links and minimizing the external links. The multi-objective optimization problem is composed of multiple objective functions and some related equality and inequality constraints. The solutions are obtained through the use of Pareto optimality theory [3] and constitute global optimum solutions satisfying all the objectives as best as possible. The evolutionary algorithm for solving multi-objective optimization problems is successful because of their population-based nature, which allows the simultaneous production of multiple optima and a good approximation of the Pareto front [4]. Community detection could be formulated as a multi-objective optimization problem, and the framework of Pareto optimality can provide a set of solutions corresponding to the best compromise among the optimization objectives.

Many approaches have been proposed to employ multi-objective techniques for data clustering. Most of these approaches cluster objects in metric spaces [5, 6, 7], though a method for partitioning graphs has been presented in [8] and a graph clustering algorithm of web user sessions is described in [9].

In recent years, researchers have gradually tended to use artificial intelligence technology to optimize modularity to find the ideal community structure. The intelligent optimization algorithms imitate natural phenomena and have a long-term observation, practice, and a profound understanding of natural phenomena. Such as intelligent optimization algorithms imitate human thinking, biological behavior, and physical principles. They all start from the stochastic feasible initial solution and approach the optimal solution of the problem through the strategy of eliminating the fittest. Although these intelligent optimization algorithms cannot guarantee that the optimal solution of the problem can be obtained eventually, they can achieve a certain balance between computational complexity and search accuracy. Until now, many intelligent optimization algorithms have been proposed, such as ant colony algorithm [10], particle swarm algorithm [11], genetic algorithm [12], differential evolution algorithms [13], etc. The complex network community detection algorithm proposed in this paper is to detect community structure. The problem is transformed into a multi-objective optimization subproblem, and the optimal global solution or a series of complementary dominant solutions are obtained through multiple objective functions of an adaptive fast genetic algorithm [12]. These solutions correspond to the community structure of complex networks. Our main contributions can be summarized as follows:

- We design two objective functions. Compared with the single objective method, the advantages of the multi-objective method are that it can optimize the multiple criteria simultaneously, and provide a set of solutions instead of a single solution. Each solution corresponds to a different number of communities, so as to find the best equilibrium.

- We design an external elite gene pool to store non-inferior solutions with high fitness. For the duplicate individuals that already exist in the elite gene pool, a series of processes such as decoding and calculating the fitness of individuals can be avoided. Due to the convergence of the solution set, the introduction of the elite gene pool reduces the computational complexity to the greatest extent.

- Logistic adaptive mutation probability and crossover probability are introduced into the algorithm. The mutation probability and crossover probability are changed according to the fitness of the population and the correlation characteristics between individuals. The efficiency and accuracy of the evolutionary process of the genetic algorithm can be significantly improved.
- Extensive experiments on several datasets demonstrate that our proposed method produces significantly increased performance over the current state-of-the-art methods in most cases.

2. Related Work

The evolutionary computation is a powerful search and optimization technology inspired by the natural evolutionary process [14]. Compared with traditional calculus-based and exhaustive optimization methods, evolutionary computation is a mature global optimization method with high robustness and wide applicability. It has the characteristics of self-organization, self-adaptation, and self-learning. It can deal with complex problems which are challenging to be solved by traditional optimization algorithm without the restriction of the nature of the problem. These methods include population initialization followed by mutation and selection operators to improve the standard values. When exploring the search space in the optimization process, the local minimum can be avoided. Many heuristic search algorithms have been applied to solve the optimization problem. The extremal optimization method, applied by Duch and Arenas, uses the artificial intelligence method in a recursive divisive manner [15]. The simulated annealing is used to obtain more results, but this method is computationally very expensive.

In addition, the genetic algorithm, as an effective optimization technique, has also been used to optimize Q [16] value. However, inefficient genetic representation makes the algorithm unsuitable for large scale problems. In fact, Arenas, Fernadez, and Gomez introduced the tabu heuristic to optimization the modularity, which also obtained an excellent performance [17].

Over the past decade, many researchers have applied evolutionary algorithms to community detection. Recent advances can be found in the literature [18, 19, 12]. Finding community structure in a network can be regarded as a clustering analysis problem. Clustering is also an optimization problem. In [20], the authors proposed a memory algorithm for community detection through module optimization and used multi-level learning strategies based on the node level, community level, and network partition level to accelerate the optimization process. The authors [13] used a combination of genetic algorithm and distance measurement based on a random walk to find subgroups in social networks. In [21], the authors proposed a discrete framework for particle swarm optimization (PSO). Based on the discrete framework, a multi-objective discrete particle swarm optimization algorithm is proposed to solve the network clustering problem. Zadeh and Kobti [22] proposed an evolutionary algorithm based on knowledge and used a multi-population cultural algorithm to solve the problem of community detection. The algorithm mainly extracts knowledge from the network to guide the search direction and find the optimal solution. At the same time, in each step, the knowledge is updated according to the current state of the network. However, these algorithms have some shortcomings. Ma, Lijia, et al. [20] proposed a sub-community of the MLCD algorithm, which is the basic unit of merging and splitting. However, if a vertex is misclassified as a sub-community, it is difficult to jump out of the sub-community at a later stage. In [13], the proposed algorithm can only solve the problem of community detection when the number of communities is known.

There are some shortcomings in the application of traditional genetic algorithms. Because the traditional genetic algorithm uses the method of fixed strategy parameters, it cannot meet the requirements of dynamic and changing strategy parameters in the evolutionary process, especially the crossover probability and mutation probability, which causing the optimization effect is not ideal. For biological evolution, even if the traditional genetic algorithm takes into account the simulation of the population's adaptability to the environment, it ignores the adaptive characteristics of genetic behavior and individual growth. Especially when the population follows the evolution of the environment, it causes a fundamental reason affecting the performance and efficiency of traditional genetic algorithms.

In the early stage of population evolution, the genetic operators of the adaptive fast genetic algorithm should be searched on a large scale to avoid premature convergence. In the later stage of population evolution, the population should be searched locally, and the evolutionary strategy should be adjusted to evolving in the critical direction. The improvement of the adaptive fast genetic algorithm in this paper mainly includes the following two points.

1. According to the fitness and similarity coefficients of individuals, the adjustment formulas of F-SGCD's adaptive crossover probability [12] and mutation probability are designed to improve the optimization ability of the algorithm.
2. The elite gene pool is introduced to store individuals with high adaptability in the evolutionary process of the genetic algorithm. In the iteration process, for the duplicate individuals already existing in the elite gene pool, a series of processes such as decoding and calculating the fitness of individuals can be avoided, which improves the efficiency of the algorithm.

3. Algorithm Description

In this section, We give a detail description of the multi-objective algorithm F-SGCD, which is used to partition the network. Many people involved in the research area adopted for the development of multi-objective optimization algorithms using the application of evolutionary computation in recent years. Evolutionary algorithms results [23] show that the evolutionary algorithm is a feasible and effective solution to multi-objective optimization problems. It is a very successful type of algorithms that it is population-based and allows the generation of several elements of the Pareto set in a single run. The detailed description of the algorithm is shown in Figure 1.

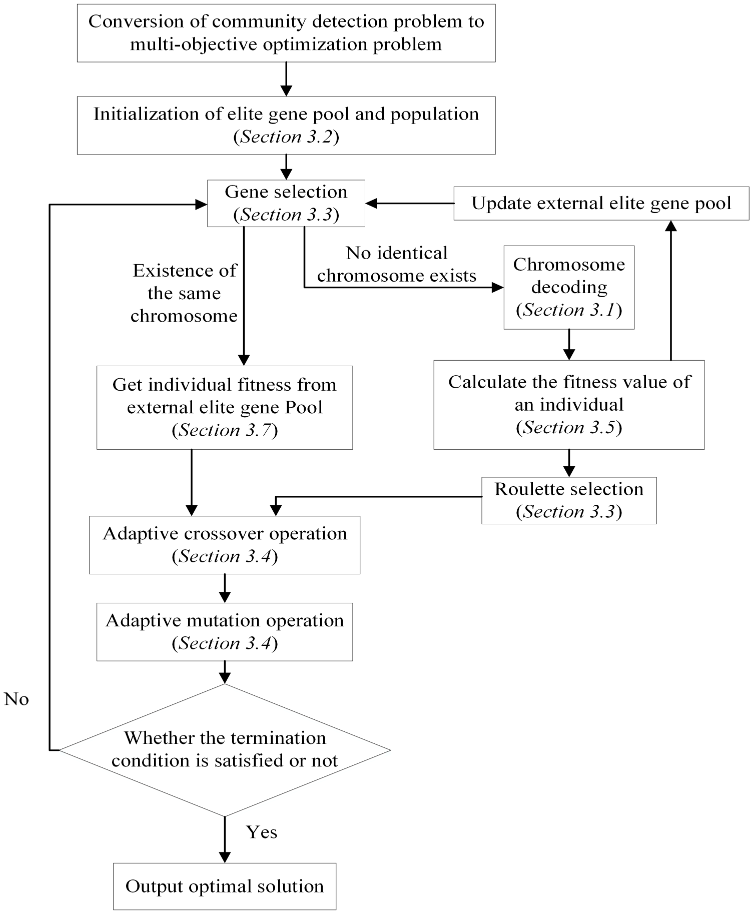

Conversion of community detection problem to multi-objective optimization problem
Initialization of elite gene pool and population (Section 3.2)
Gene selection (Section 3.3)
Update external elite gene pool
Existence of the same chromosome
No identical chromosome exists
Chromosome decoding (Section 3.1)
Get individual fitness from external elite gene Pool (Section 3.7)
Calculate the fitness value of an individual (Section 3.5)
Roulette selection (Section 3.3)
Adaptive crossover operation (Section 3.4)
Adaptive mutation operation (Section 3.4)
No
Whether the termination condition is satisfied or not
Yes
Output optimal solution

Figure 1: Flow chart of F-SGCD algorithm.

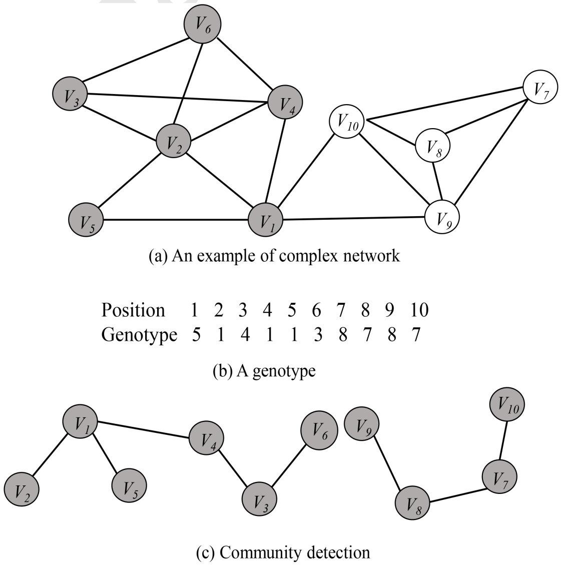
3.1. Encoding Scheme of F-SGCD

We adopt a coding method based on the adjacency representation of genes
150 [24, 25]. In this encoding style, individuals in the population are composed of N genomes $\{g_1, g_2, \ldots, g_N\}$, and the range of allele value for each gene is $\{1,2,\ldots,N\}$. Both genes and alleles represent the vertices of G. For example, if the allele value of the i-th gene is j, which is expressed as there is an edge between

vertices i and j. The encoding method needs a decoding step to identify each connected subnetwork, and vertices located in the same connected subnetwork are divided into the same community.

The advantage of the encoding method is that it does not need to know the number of communities in advance, and the number of communities can be calculated automatically during the decoding process. Figure 2 shows an example based on the adjacency representation of genes. Figure 2 (a) shows a complex network composed of 10 vertices, in which the solid and hollow circles represent vertices of two communities, respectively. Figure 2 (b) shows the genotype of an individual. Figure 2 (c) is the decoding results of the community according to the individual encoding in Figure 2 (b). According to the calculation formula of the modularity Q (Section 3.6), The Q value of partitioning result for the example network in Figure 2 is as follows.

$$Q = \frac{11}{19} - \left(\frac{2}{19}\right)^2 + \frac{7}{19} - \left(\frac{2}{19}\right)^2 = 0.8725 \tag{1}$$



(a) An example of complex network

Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	4	1	1	3	8	7	8	7

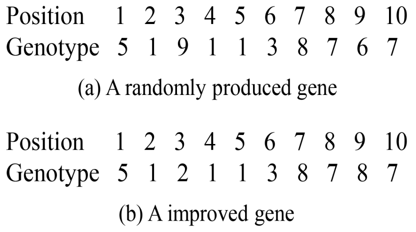
(b) A genotype

(c) Community detection

Figure 2: An example of encoding and decoding process.

3.2. Initialization

Firstly, the population includes several individuals who are generated randomly. In the encoding method based on the adjacency representation of genes described in the previous section, if the allele of the gene i is j, it means that there is an edge between vertices i and j. In the output results after decoding, the vertices i and j should be in the same community. Therefore, in the initialization process of the optimization algorithm, we distribute it reasonably according to the connection relationship of the vertices. That is to say, for the initialization of each individual, the allele value j of gene i can only be the adjacent vertex of i. if the allele value of the i-th gene is j, but the edge (i, j) does not actually exist, then the allele value j will be replaced by an adjacent vertex of i. For example, in Figure 3 (a), the allele values of the third and ninth genes are 9 and 6, respectively. However, the edges of (3, 9) and (9, 6) do not appear in the network of Figure 3 (a). Therefore, the allele value 9 of the third gene can be replaced by 2, and the allele value 6 of the ninth gene can be replaced by 8. The replaced gene is shown in Figure 3 (b). The initialization method can effectively restrict the size of the solution space, greatly reduce the invalid search in the evolution process of the algorithm, and significantly improve the convergence speed of the algorithm.



Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	9	1	1	3	8	7	6	7

(a) A randomly produced gene

Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	2	1	1	3	8	7	8	7

(b) A improved gene

Figure 3: Improvement diagram of gene initialization.

3.3. Selection

The purpose of selection is to eliminate unreasonable individuals according to the fitness of individuals and the principle of survival of the fittest. F-

SGCD uses the roulette selection method to achieve ”survival of the fittest” among individuals in the current population, which is similar to roulette in gambling games. In the process of roulette selection, the probability of each individual inherited into the next generation is equal to the ratio of its fitness value to the sum of individual fitness values in the whole population. The higher the fitness value of the individual, the higher the probability of the individual being selected, and the greater the probability of being inherited to the next generation. Assuming that the population size is M and the fitness value of individual k is f_k. The probability of each individual being inherited into the next generation is,

$$p_k = \frac{f_k}{\sum_{i=1}^{M} f_i},\ k = 1, 2, ..., M \tag{2}$$

The cumulative probability of each individual is,

$$q_k = \sum_{i=1}^{k} p_i,\ k = 1, 2, ..., M \tag{3}$$

The selection process is to rotate the runner M times, according to the following steps to select an individual to join the new population at every time.

Step 1. Generate a uniformly distributed pseudo-random number r in interval [0,1];

Step 2. If $r \leq q_1$, select the first individual. Otherwise, select the k-th $(2 \leq k \leq M)$ individual to make r $(q_{k-1} \leq r \leq q_k)$ valid;

Step 3. Repeat the above steps M times.

3.4. Uniform Crossover and Mutation

Individuals with poor fitness should be given smaller mutation probability and larger crossover probability. For individuals with better fitness, the crossover probability and mutation probability are given according to the size of the fitness value of the individual and the iterative state of the population. The closer the number of iterations approaches the maximum number of iterations, the smaller the crossover probability of individual and the larger the mutation probability, so that the population evolution will not be stuck in a stagnant state.

In each iteration process, when the similarity between individuals is small, the fitness values between individuals vary considerably, which indicates that there are abundant genotypes in this population. Therefore, a larger crossover probability and a smaller mutation probability should be given. If the similarity between individuals is large and there is little difference in fitness values between individuals, it indicates that there are fewer genotypes in the population. Therefore, A smaller crossover probability and a larger mutation probability should be given.

In order to search the solution space more effectively, the concepts of standard deviation and similar parameters are introduced. Standard deviation is a measure of the average dispersion of a set of data. A large standard deviation represents a large difference between most values and their average value. A smaller standard deviation means that these values are closer to the average. The similarity parameter reflects the similarity degree of individuals in the current population. When the similarity parameter is large, it shows that the similarity degree of individuals is high, the algorithm tends to converge, and the overall performance of individuals is excellent. On the contrary, it shows that the similarity degree of individuals is low, and the overall performance of the population is poor.

$$g_{avg} = \frac{g_1 + g_2 + ... + g_N}{N} \tag{4}$$

$$\sigma = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N}(g_i - g_{avg})^2\right)} \tag{5}$$

$$\varOmega = \frac{g_{avg} + 1}{\delta} \tag{6}$$

where N denotes the number of individuals in the population. g_1,g_2,...,g_N denotes the fitness of the individual. g_{avg} is the arithmetic average of population fitness, which reflects the average fitness of individuals. σ denotes standard deviation. Ω represents similarity parameter. With the increase of generations number, the average fitness of population is higher and higher, but the standard deviation is smaller and smaller, and the value of similarity parameter is larger and larger.

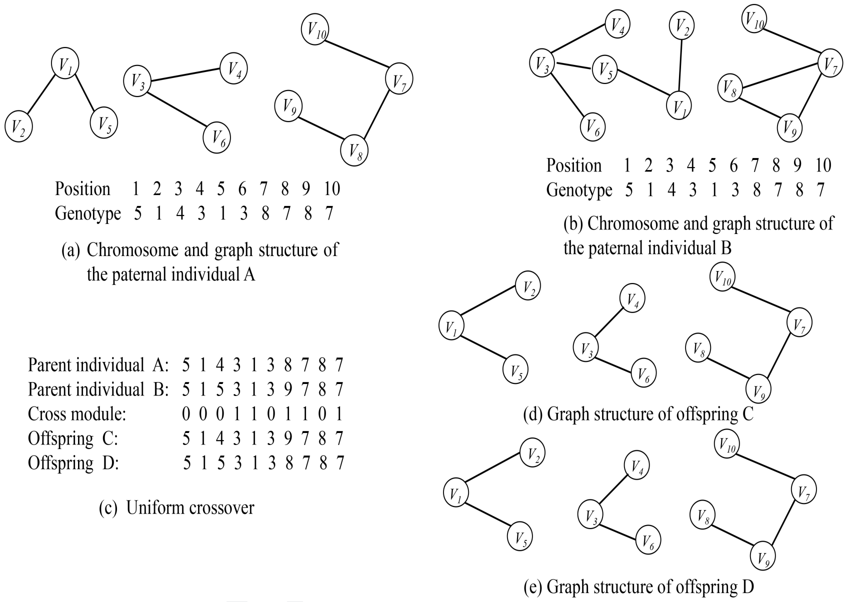
According to the above design criteria of crossover probability and mutation probability, the concept of standard deviation and the definition of related parameters are combined. the dynamic adjustment formulas of crossover probability p_c and mutation probability p_m can be designed as follows:

$$p_c = 0.5 \times \frac{1}{1 + e^{-\frac{k_1}{\Omega}}} + 0.4 \tag{7}$$

$$p_m = \frac{k_2}{5 \times \left(1 + e^{\frac{1}{\Omega}}\right)} \tag{8}$$

where k_1 and k_2 are constants, $k_1 \in (1,\ \infty)$, $k_2 \in (0,\ 1)$. It can be seen from the adaptive adjustment formulas of crossover probability and mutation probability, $p_c \in (0.65,\ 0.9)$, $p_m \in (0,\ 0.1)$. The values of crossover probability and mutation probability are within a reasonable range. As the square difference increases, the crossover probability decreases and the mutation probability increases, so the crossover probability and mutation probability meet the two design criteria of crossover probability and mutation probability.

An improved uniform crossover operator is adopted in F-SGCD to ensure the effectiveness of the offspring. The crossover probability of uniform crossover for the individual population is p_c, and the crossover operation is carried out at each position of the chromosome of the paternal individual with the same probability. Firstly, a binary cross-module of length N (number of vertices) is generated randomly. Each value on the cross-module is 0 or 1. For each gene of offspring C, if a position on the cross-module is 1, the corresponding allele value in the parent B is inherited. If a position on the cross-module is 0, the corresponding allele value in the parent A is inherited, while the opposite is true for the offspring D. in practice, the preferred initialization mentioned in the previous section is used, i.e. the i-th gene in the parent has an allele value j, then the edge (i, j) will exist. By means of uniform crossover, the value of each gene location in the offspring is inherited from the parent, which can ensure the effective connection of each node in the network of offspring individuals.



Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	4	3	1	3	8	7	8	7

(a) Chromosome and graph structure of the paternal individual A

Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	4	3	1	3	8	7	8	7

(b) Chromosome and graph structure of the paternal individual B

Parent individual A:	5 1 4 3 1 3 8 7 8 7
Parent individual B:	5 1 5 3 1 3 9 7 8 7
Cross module:	0 0 0 1 1 0 1 1 0 1
Offspring C:	5 1 4 3 1 3 9 7 8 7
Offspring D:	5 1 5 3 1 3 8 7 8 7

(c) Uniform crossover

(d) Graph structure of offspring C

(e) Graph structure of offspring D

Figure 4: An example of uniform crossover.

The mutation operation is as follows. For the individual to be mutated, a gene is randomly selected by adaptive mutation probability p_m, and the allele value of the gene is changed to its corresponding arbitrary adjacent vertex. This mutation method also avoids searching for invalid solution space. Therefore, the possible value of an allele is limited to the adjacent gene of gene i. For example, in the network topology of Figure 3 (a), the allele value of the gene at the third position is only 2,4,5,6.

In the early stage of evolution, mutation operation is carried out, and the mutation probability in the process of evolution is set to the adaptive mutation probability p_m mentioned above. If the mutation operator randomly changes the allele value j of gene i, it will lead to an invalid search of search space, so in the actual mutation process, the allele value of the gene changes to its corresponding arbitrary neighbor nodes. This mutation method ensures that each vertex is connected to only one of the adjacent vertices in the generated mutation offspring, and improves the search efficiency of the solution space.

3.5. Fitness function

Establishing an objective function is the most crucial step in optimal design. With the deepening of research on complex networks, it is found that the nature of community structure is often found in many complex networks, which is characterized by relatively close links within communities and relatively sparse links among different communities. This partitioning has two competing objectives: one is to minimize the links between communities, the other is to maximize the links with the vertices in the community. The problem of community detection should not be neglected in order to meet one goal. Therefore, it is suitable for multi-objective optimization.

In the study of community detection for complex networks, we construct two objective functions, namely community fitness function and community score function. The first objective function is to minimize community fitness value, which is expressed as follows.

$$fitness = \sum_{j=1}^{k} p\left(S_j\right) \tag{9}$$

$$P\left(S_j\right) = \sum_{i \in s} \frac{k_i^{in}\left(S\right)}{k_i^{in}\left(S\right) + k_i^{out}\left(S\right)} \tag{10}$$

where k is the number of communities, ${k_i}^{in}(S)$ denotes the number of edges of the vertex i connects the other vertices in subnetwork S and ${k_i}^{out}(S)$ represents the number of edges of the vertex i connects the other vertices outside the subnetwork S. The second objective function is to maximize the community fitness value, which is expressed as:

$$CS = \sum_{i=1}^{k} score\left(S_i\right) \tag{11}$$

$$score\left(S\right) = M\left(S\right) \times E_s \tag{12}$$

$$M\left(S\right) = \frac{\sum_{i \in s} \left(u_i\right)^r}{|S|} \tag{13}$$

$$u_i = \frac{1}{|S|} k_i^{in}\left(S\right) \tag{14}$$

where $|S|$ denotes the number of vertices in community S. E_s denotes the total number of edges in community S, and u_i denotes the proportion of the number of neighbors of vertex i in S to the number of all vertices. r is called a resolution parameter and is a positive real number. Generally, it is set to 2 and used to control the size of the network community. Because $0 \leq u_i \leq 1$, $r \geq 1$, the weights of vertices with more connections in community S are strengthened, and those with less connections in community S are weakened. For a complex community network, when ${k_i}^{out}(S)$=0, $P(S)$ reaches its maximum value.

3.6. The solution selection of Pareto

Each solution of the F-SGCD algorithm represents different trade-offs between two objectives, which results in many different network community detection schemes. Therefore, we need to establish a standard parameter to select a Pareto solution. The multi-objective adaptive fast genetic algorithm adopts modularity proposed by Girvan and Newman [26].

$$Q = \sum_{s=1}^{k} \left[\frac{l_s}{m} - \left(\frac{d_s}{2m} \right)^2 \right] \tag{15}$$

l_s denotes the total number of edges in community S, d_s is the sum of degrees of vertices in S, and m is the total number of edges in the network. The closer the Q value to 1, the stronger the community structure. In the actual network, the value usually ranges from 0.3 to 0.7. In the fast multi-objective adaptive genetic algorithm, the Pareto solution set formed by multi-objective partition is stored in the elite gene bank. In this paper, modularity is regarded as an optimal solution selection criterion, and the network corresponding to the solution with the largest modularity is divided into the optimal partition of the current network.

3.7. Elite gene pool

After crossover, mutation, and selection, the squared difference of the improved genetic algorithm decreases gradually. The individuals tend to converge, and the individuals with the same gene coding will increase progressively. Therefore, we introduce the elite gene pool to store those individuals with higher adaptability. For the repetitive individuals that already exist in the elite gene pool, a series of processes such as decoding and calculating individual fitness can be avoided. The elite gene pool can reduce the complexity of the algorithm, improve the efficiency and practicability of the operation. The size of the elite gene pool is 0.2∼0.3 times the size of the individual population.

According to the fitness of chromosomes in the current population, the coding of chromosomes with higher fitness and the corresponding fitness values are added to the elite gene pool, and the fitness values are sorted from large to small. Then, In the evolutionary process of offspring, when calculating the individual fitness, we first search for individuals with the same coding in the elite gene pool. If the same individual exists, the fitness value of the corresponding individual in the elite gene pool is directly taken as the fitness value of the current individual; if not, the fitness value is calculated according to the fitness function, and the fitness value is compared with that of the individuals in the elite gene pool, if the fitness value is greater than the minimum fitness in the elite gene pool, the individual is put into the elite gene pool. If the number of individuals in the elite gene pool reaches the specified size, the less fitness individual in the elite gene pool is discarded.

In the prophase of the genetic algorithm, the individuals in the population change substantially, the individual repetition rate is relatively low. However, after a certain generation number, the elite gene pool will be constructed to reduce the inefficiency of the elite gene pool in the early stage.

Although the introduction of the adaptive fast genetic algorithm into the elite gene pool in the actual solution, it is more complex than the traditional adaptive genetic algorithm. The elite gene pool is, to the greatest extent, to avoid the individual's adaptability to the problem of repeated calculation. By the late stage of evolution, the role of the elite gene pool became more and more obvious as the repetitive individuals appeared frequently.

4. The algorithm flow of F-SGCD

In the F-SGCD algorithm, the first step is to initialize the population randomly. Each individual represents a network structure, and each of its components is a connected subgraph of G. F-SGCD calculates the two objective function values of each individual, ordering and classifying the two objective function values of each individual according to the domination between Pareto solutions, and then executes an adaptive crossover mutation operator to generate a new population. After several iterations, the F-SGCD algorithm finally returns a Pareto optimal solution with the highest modularity. The procedure of F-SGCD is summarized in Algorithm 1.

In the framework of algorithm 1, *Initialization*() is used to initialize the population. *Select*() is a selection operation in F-SGCD algorithm. *Adaptive*() is used to calculate adaptive crossover probability p_c and adaptive mutation probability p_m according to Formula (7) and Formula (8). the functions *Mutation*() and *Crossover*() represent mutation operation and crossover operation, respectively. *Update*() represents updating the current population, i.e. selecting individuals with higher fitness from population P and P_{child}. *Termination*() denotes the termination condition of the loop statement. *ElitePool*() is to update the genes in the elite gene pool, which ranks non-Pareto non-inferior solutions according to fitness values. Given a graph G=(V, E), F-SGCD performs the

Algorithm 1 Framework of F-SGCD Algorithm

Input: Population size *Population*; Maximum number of iterations *Generation*

Output: The results of community detection

```
Initialization: Adaptive parameters: adaptive crossover probability $p_c$ of population $P$, adaptive mutation probability $p_m$ of population $P$
$P \leftarrow Intialization(Population)$;
While $Termination(Generation)$;
$P_{parent} \leftarrow Select(P)$
$p_c$, $p_m \leftarrow Adaptive()$;
$p_{cross} \leftarrow Crossover(P_{parent}, p_c)$
$p_{child} \leftarrow Mutation(P_{cross}, p_m)$
$P \leftarrow Update(P_{child})$;
$ElitePool()$;//{Update the elite gene pool}
End;
return The results of community detection //{Transforming the most adaptable non-inferior solutions from the elite gene pool into community detection results}
```

following specific steps:

Step 1. The community detection problem is transformed into a multi-objective problem, and two objective functions are established, namely the community score objective function and the community fitness objective function.

Step 2. According to the adjacency principle, the individuals in the population and the elite gene pool are initialized, and the size of the elite gene pool is about 0.2∼0.3 times the size of the population.

Step 3. For gene selection, if the same chromosome exists in the elite gene pool, the fitness of individuals can be obtained from the gene pool. If the same chromosome does not exist, the chromosome is decoded. According to the domination between the two objective functions, the elite gene pool is updated by sorting according to the Q value.

Step 4. Select individuals by roulette.

Step 5. Calculate the adaptive crossover probability and mutation probability, then perform a crossover mutation operation to generate the next generation population.

Step 6. Determine whether the maximum number of iterations is reached. If so, the non-inferior solution with the highest modularity in the elite gene pool is returned as the optimal solution of the final output.

5. Experimental results and discussions

To test the performance of the F-SGCD algorithm, real-word networks, and artificial networks of different scales are used to carry out comparative experiments. Furthermore, we also applied the same networks in the competing algorithms, including CNM [26], SCORE [27], LPA [28], MOEA [29], SPOC [30] and FuzAg [31]. The experiment was carried out on a single computer with 3.1GHz Pentium 4CPU and 16GB memory. The software platform is python 2.7 in Windows.

5.1. Datasets

5.1.1. Real-word networks

We chose four real-world networks, including Bottlenose Dolphins [32], Zachary's Karate Club [32], American Coll. Football [33] as well as Krebs' book [33]. Table 1 describes in detail the characteristics of four network structures (the number of edges (m), the number of nodes (n), and the number of real communities ($|C|$)).

Networks	n	m	$\|C\|$	Ground truth
Karate	78	34	2	Known
Dolphin	159	62	2	Known
Football	613	115	12	Known
Krebs' Books	441	105	3	Known

Table 1: Characteristics of four network structures.

5.1.2. Artificial networks

We use the LFR (Lancichinetti-Fortunato-Radicchi) Benchmark [34] proposed by Lancichinetti and Fortunato to produce artificial networks, which to test the feasibility and validity of the algorithm. The network consists of 128 vertices and 4 communities. Each community has 32 vertices, and the average degree of vertices is 16. The mixed parameter μ controls the ratio of the external degree of the vertex to the vertex degree. The smaller the value μ, the smaller the proportion of vertices connected to other communities, and the clearer the community structure. In the experiment, we adjusted the value of μ from 0 to 0.5, and the interval is 0.05. When $\mu \doteq 0.5$, half of the vertices connected with each vertex are in other communities, and the community structure is relatively vague. When $\mu < 0.5$, the ratio of the external degree of a vertex is less than that of internal degree. When $\mu \doteq 0$, the ratio of the external degree of vertex to the vertex degree is 0, and the vertex is only connected with the vertices in their own community. At this time, the community structure is the most obvious.

5.2. Evaluation metrics

Normalized Mutual Information (NMI). *NMI* is a useful information measure in information theory. It is introduced by Leon Danon et al. [35] and used to measure the similarity between the detected communities and the known communities. Given two partitions A and B of a network in communities, let C be the confusion matrix whose element C_{ij} is the number of nodes of community i of the partition A that are also in the community j of the partition B. The normalized mutual information $I(A,\, B)$ is defined as follows:

$$I\left(A,B\right)=\frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}C_{ij}\log\left(C_{ij}N/C_{i\cdot}.C_{\cdot j}\right)}{\sum_{i=1}^{C_A}C_{i\cdot}\log\left(C_{i\cdot}/N\right)+\sum_{j=1}^{C_B}C_{\cdot j}\log\left(C_{\cdot j}/N\right)} \tag{16}$$

where C_A is the number of groups in the partition A, C_B is the number of groups in the partition B. $C_{i\cdot}$ is the sum of the elements of C in row i. $C_{\cdot j}$ is the sum of the elements of C in column j. N denotes the number of nodes. The closer the *NMI* value is to 1, the more similar the detected community is to the known community. On the contrary, the closer the *NMI* value is to 0, the less similar the detected community is to the known community.

Modularity. Newman et al. citeclauset2004finding proposed the concept of modularity, which is a criterion for evaluating the quality of community detection. The definition of modularity is as follows:

$$Q = \sum_{C} \left[\frac{l_C}{m} - \left(\frac{k_C}{2m} \right)^2 \right] \tag{17}$$

where l_C the total number of edges joining vertices of community C and k_C the sum of the degrees of the vertices of C. Large values of Q are then supposed to indicate partitions with high quality.

5.3. Experiments on synthetic LFR networks

Figure 5 shows that the number of iterations of the F-SGCD algorithm is set to 100 times, and the population size is set to be different. Under the Benchmark simulation network with different mixed parameter μ, the algorithm runs 20 times, taking the maximum value of NMI as a result. As can be seen from Figure 5, when $\mu \leq 0.25$, the maximum NMI values of the F-SGCD algorithm with population sizes of 50, 100, 150, and 200 are all 1, which shows that the effect of community detection is perfect. When $0.25 \leq \mu \leq 0.3$, the NMI values of population size 50 and 100 began to decrease, while the values of other populations remained at 1. When $\mu \geq 0.3$, the NMI value of population size 150 began to decline, while the NMI value of population size 200 remained at 1, and the NMI value of population size 50 and 100 continued to decline. When $0.35 \leq \mu \leq 0.4$, the NMI of the F-SGCD algorithm with population size 100 and 150 decreases faster, while the maximum value of NMI with population size 200 is still 1. When $\mu > 0.45$, the NMI values of the F-SGCD algorithm with a population size of 200 decreases sharply, while those of 50, 100, and 150 decreases slowly. When r decreases to 0.5, the NMI values of 50, 100, 150, and 200 are similar.

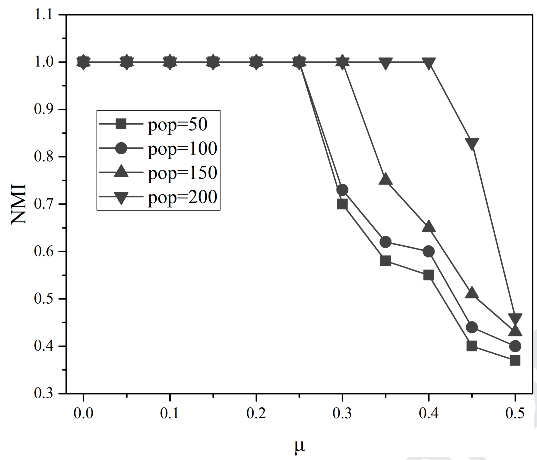


Figure 5: The maximum NMI value obtained by F-SGCD algorithm running Benchmark network 20 times under different population sizes.

Figure 7 shows that the population size of the F-SGCD algorithm is set to 150, and the maximum number of iterations is set to be different. In the Benchmark simulation network with different mixed parameters, the algorithm runs 20 times, taking the maximum value of NMI as a result. We can see from Figure 5 that when $\mu \leq 0.25$, the maximum NMI of the F-SGCD algorithm is 1 when the number of iterations is 50, 100, 150 and 200, which shows that the effect of community detection is perfect. When $\mu \doteq 0.25$ iteration times are 50, the NMI of iteration times 100, 150, and 200 are 1, when $\mu \geq 0.45$, the maximum NMI values decrease at the same rate for different iterations.

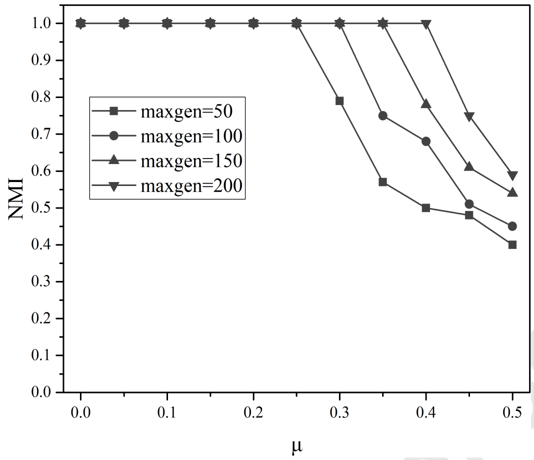


Figure 6: The maximum NMI value obtained by F-SGCD algorithm running Benchmark network 20 times under different iterative times.

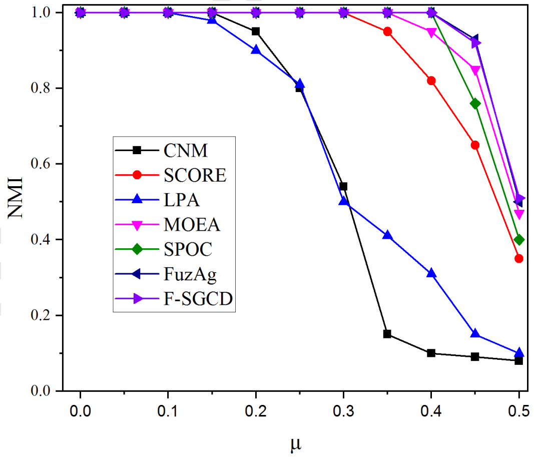


Figure 7: Comparison of F-SGCD and other algorithms.

5.4. Experiments on real-world networks

Table 2 shows the results of 20 runs of different algorithms in four real 475 networks. NMI_{max} denotes the maximum of 20 results, NMI_{avg} denotes the average of 20 results. FuzAg, F-SGCD, SPOC, and MOEA algorithms have a better effect. Among them, many values of NMI_{max} and NMI_{avg} are 1. The results of CNM, SCORE, and LPA algorithms are poor. For the Karate network, the NMI_{avg} values of the F-SGCD algorithm are 9.1%, 5.5%, and 15.2% higher than those of CNM, SCORE, and LPA, respectively. For the Dolphin network, the NMI_{avg} values of the F-SGCD algorithm are 14.4%, 1.9%, and 5.0% higher than those of CNM, SCORE, and LPA, respectively. For the Football network, the NMI_{avg} values of the F-SGCD algorithm are 14.4%, 1.9%, and 5.0% higher than those of CNM, SCORE, and LPA, respectively. For Krebs' books network, the NMI_{avg} values of the F-SGCD algorithm are 26.7%, 4.7%, and 36.5% higher than those of CNM, SCORE, and LPA, respectively. LPA algorithm is very unstable because its NMI_{max} is quite different from NMI_{avg}.

Network	Karate		Dolphin	
	NMI_{avg}	NMI_{max}	NMI_{avg}	NMI_{max}
CNM	0.890	0.963	0.872	0.968
SCORE	0.942	1.000	0.901	0.972
LPA	0.973	0.981	0.826	0.957
MOEA	0.956	1.000	0.912	1.000
SPOC	0.857	1.000	0.943	1.000
FuzAg	1.000	1.000	0.945	0.991
F-SGCD	**1.000**	**1.000**	**0.951**	**1.000**
Network	Football		Krebs' books	
	NMI_{avg}	NMI_{max}	NMI_{avg}	NMI_{max}
CNM	0.817	0.894	0.419	0.472
SCORE	0.918	1.000	0.507	0.581
LPA	0.891	1.000	0.389	0.417
MOEA	0.925	0.981	0.498	0.590
SPOC	0.896	0.904	0.521	0.530
FuzAg	0.924	1.000	0.527	0.597
F-SGCD	**0.935**	**1.000**	**0.531**	**0.610**

Table 2: *NMI* values of the eight compared algorithms on four real-world networks, averaging over 20 runs.

Table 3 shows the results of 20 runs of different algorithms in four real networks. Q_{max} denotes the maximum of 20 results, Q_{avg} denotes the average of 20 results. FuzAg, F-SGCD, SPOC, and MOEA algorithms have better effect. Especially for FuzAg, the Q_{max} value in the result of Karate exceeds the algorithm proposed in this paper. The results of CNM, SCORE, and LPA algorithms are poor. For the Karate network, the Q_{avg} values of the F-SGCD algorithm are 15%, 8.7%, and 33.9% higher than those of CNM, SCORE, and LPA, respectively. For the Dolphin network, the Q_{avg} values of the F-SGCD algorithm are 8.7%, 32.1%, and 141% higher than those of CNM, SCORE, and

LPA, respectively. For the Football network, the Q_{avg} values of the F-SGCD algorithm are 13.5%, 8.8%, and 29.3% higher than those of CNM, SCORE, and LPA, respectively. For Krebs' books network, the Q_{avg} values of the F-SGCD algorithm are 11.2%, 14.2%, and 238.3% higher than those of CNM, SCORE, and LPA, respectively. LPA algorithm is very unstable because its Q_{max} is quite different from Q_{avg}.

Overall, for F-SGCD, the performance of finding optimal solutions has been improved in the community detection process. It can produce results that are close to the real network and exhibit better performance.

Network	Karate		Dolphin		Football		Krebs' books	
	Q_{avg}	Q_{max}	Q_{avg}	Q_{max}	Q_{avg}	Q_{max}	Q_{avg}	Q_{max}
CNM	0.380	0.380	0.480	0.501	0.591	0.609	0.493	0.513
SCORE	0.402	0.402	0.429	0.473	0.617	0.654	0.480	0.480
LPA	0.326	0.415	0.217	0.396	0.519	0.623	0.162	0.276
MOEA	0.419	0.421	0.487	0.501	0.604	0.647	0.483	0.517
SPOC	0.416	0.419	0.521	0.525	0.601	0.604	0.506	0.523
FuzAg	0.423	0.480	0.517	0.520	0.627	0.681	0.516	0.516
F-SGCD	**0.437**	0.479	**0.522**	**0.526**	0.671	**0.694**	**0.548**	**0.529**

Table 3: Q values of the eight compared algorithms on four real-world networks, averaging over 20 runs.

5.5. Network hierarchy of Pareto solution

Modules [36] refer to a group of nodes that are physically or functionally linked together to accomplish a relatively independent function. Many systems contain modules, and high modularity is the basic design requirement of a large complex system. The modules of the network are identified according to the network topology, and the effectiveness of module partition can be explained by analyzing the relationship between these modules and functions. In highly connected community sub-networks, nodes with a small degree have high clustering coefficients; On the contrary, high degree central nodes have lower clustering coefficients, which only connect different subnetworks. In a hierarchical modular network, many small-scale nodes with dense internal connections are relatively loose, thus forming a larger-scale topology module. This kind of topology structure is arranged hierarchically, and the network generated by module iteration is called a hierarchical network. Hierarchical networks have the characteristics of local clustering, modularity, and scale-free topology.

Hierarchical modularity: Module refers to a group of vertices that are physically or functionally linked together to accomplish a relatively independent function. Many systems contain modules, and high modularity is the basic design requirement of a large complex system. People can identify network modules based on the network topology structure. The validity of module partition can be demonstrated by analyzing the relationship between these modules and their functions. In densely connected community sub-networks, vertices with a small degree have a higher clustering coefficient. On the contrary, high-degree vertices with a low clustering coefficient, which only connect different sub-networks. In a hierarchical modular network, many small-scale vertices with dense internal connections are loosely connected, and thus forming a larger-scale topology module. This topological structure is arranged in hierarchical order, and the network that generates modules iteratively is called hierarchical network [37]. Hierarchical networks have the characteristics of local clustering, modularity, and scale-free topology.

Bottlenose Dolphins: Lusseau et al. [38] observed the behavior of 62 dolphins over a long period. The edge connection between the two dolphins indicated that they often contacted each other. There are 159 edges in the network, forming two communities. In Lusseau's research, firstly, the whole Dolphins network is naturally divided into two groups: A and B, which correspond to female Dolphins and male Dolphins, respectively. Through further research, Lusseau found that the male Dolphins network was further divided into three community groups, and speculated that these three network groups belonged to three different matrilineal pedigrees [36].

The Dolphins network was executed 20 times, and the eight Pareto frontiers in the elite gene pool during one operation were shown in Figure 8. The abscissa and ordinate coordinates represent the function values of community fitness and community score, respectively. The two values in the box represent the corresponding *NMI* and Q values of the solution. The maximum value of *NMI* is 1, and the maximum value of Q is 0.5196. The results of the community network partition corresponding to these two solutions are shown in Figure 9 and Figure 10. Figure 11 is the result of network partitioning when $NMI = 0.8076$. As shown in Figure 9, when $NMI = 1$, the result of partition is the same as that of real community partition, the Dolphins network is divided into two sub-networks of A and B. Figure 10 shows that the network B in Figure 9 is further divided into two sub-networks, while Figure 11 shows that the subnetwork C in Figure 10 is further divided into two more density subnetworks of C and D.

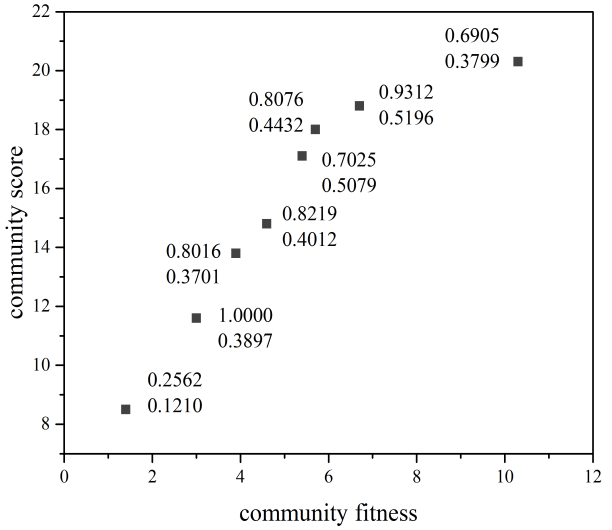


Figure 8: Pareto solution in elite gene pool when running Dolphin network.

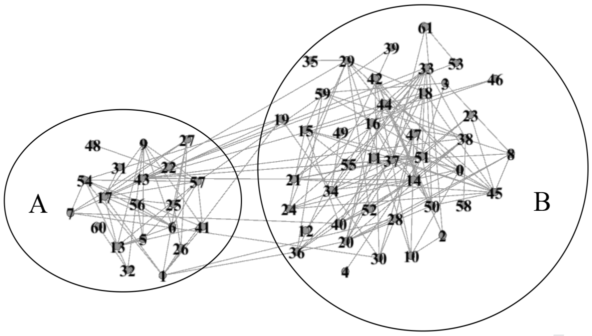


Figure 9: NMI=1, Q=0.3897.

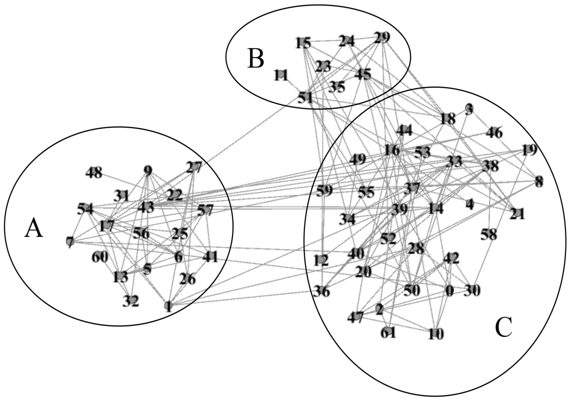


Figure 10: NMI=0.9312, Q=0.5196.

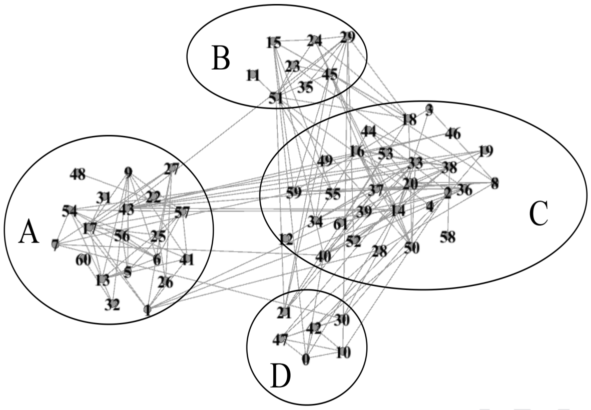


Figure 11: NMI=0.8076, Q=0.4432.

The hierarchical structure of complex networks is to integrate communities of different granularity and levels in complex networks. Some sub-networks of the complex network can also be subdivided into several density sub-communities at a low level. According to the analysis of the above experimental results, F-SGCD provides a complete set of Pareto frontier solutions for researchers in network community partitioning and finds out the hierarchical structure of the network. We can better understand the internal structure of the complex network, which is conducive to the development and utilization of the complex network. In this respect, it has more advantages than a single-objective optimization method.

6. Conclusion

The traditional community detection algorithm has the disadvantages of low efficiency and a single optimization solution. We design a multi-objective adaptive genetic algorithm. Firstly, it transforms the problem of community detection into a multi-objective optimization problem and constructs two objective functions of community score and community fitness. Secondly, an external elite gene pool is introduced to store non-inferior solutions with high fitness. For the duplicate individuals that already exist in the elite gene pool, there is no need to re-decode and re-calculate individual fitness. At the same time, an adaptive genetic operator is executed to return a set of non-dominant solutions compromised between the two objective functions. Finally, a Pareto optimal solution with the highest modularity is selected and decoded to generate a set of independent sub-networks. The performance of the algorithm is evaluated by normalized mutual information and modularity. The experimental analysis shows that the adaptive genetic operator and elite gene pool constructed according to the characteristics of the problem are helpful to improve the optimization and stability of the algorithm in community detection of complex networks. F-SGCD algorithm based on the Pareto solution is helpful in discovering the hierarchical structure of complex networks. The inherent parallel mechanism of F-SGCD and its global optimization characteristics are suitable for solving multi-objective optimization problems.

7. Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61976120, the Natural Science Foundation of Jiangsu Province under Grant BK20191445, Qing Lan Project of Jiangsu Province, and the Six Talent Peaks Project of Jiangsu Province under Grant XYDXXJS-048.

References

[1] D. J. Watts, S. H. Strogatz, Collective dynamics of 'small-world' networks, Nature.

[2] Q. Li, J. Zhong, Q. Li, Z. Cao, C. Wang, Enhancing network embedding with implicit clustering. (2019) 452–467.

[3] B. Sawik, J. Faulin, E. Pérez-Bernabeu, Multi-criteria optimization for fleet size with environmental aspects, Transportation Research Procedia 27 (2018) 61–68.

[4] L. Ke, R. Chen, G. Fu, Y. Xin, Two-archive evolutionary algorithm for constrained multiobjective optimization, IEEE Transactions on Evolutionary Computation 23 (2) (2019) 303–315.

[5] S. Rahimi, A. Abdollahpouri, P. Moradi, A multi-objective particle swarm optimization algorithm for community detection in complex networks, Swarm and evolutionary computation 39 (2017) 297–309.

[6] V. Poulin, F. Théberge, Ensemble clustering for graphs: Comparisons and applications.

[7] S. Saha, S. Bandyopadhyay, A symmetry based multiobjective clustering technique for automatic evolution of clusters, Pattern Recognition 43 (3) (2010) 738–751.

[8] D. Datta, J. R. Figueira, Graph partitioning by multi-objective real-valued metaheuristics: A comparative study, Applied Soft Computing 11 (5) (2011) 3976–3987.

[9] G. N. Demir, A. Å. Uyar, Å. Gunduzoguducu, Multiobjective evolutionary clustering of web user sessions: a case study in web page recommendation, soft computing 14 (6) (2010) 579–597.

[10] P. Ji, S. Zhang, Z. P. Zhou, A decomposition-based ant colony optimization algorithm for the multi-objective community detection (2).

[11] C. Sammut, G. I. Webb, Encyclopedia of machine learning and data mining.

[12] M. Guerrero, F. G. Montoya, R. Banos, A. Alcayde, C. Gil, Adaptive community detection in complex networks using genetic algorithms, Neurocomputing 266 (2017) 101–113.

[13] A. Firat, S. Chatterjee, M. Yilmaz, Genetic clustering of social networks using random walks, Computational Statistics δ Data Analysis 51 (12) (2007) 6285–6294.

[14] D. Fogel, Artificial intelligence through simulated evolution, in: National Conference on Emerging Trends δ Applications in Computer Science, 1966.

[15] L. Danon, A. Diazguilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment 2005 (09) (2005) 09008.

[16] M. Tasgin, H. Bingol, Community detection in complex networks using genetic algorithm.

[17] A. Arenas, A. Fernández, S. Gómez, Analysis of the structure of complex networks at different resolution levels, New Journal of Physics 10 (5) (2007) 053039.

[18] C. Pizzuti, Evolutionary computation for community detection in networks: A review, IEEE Transactions on Evolutionary Computation 22 (3) (2018) 464–483.

[19] C. Pizzuti, M. A. Smith, H. Meier, S. Kumar, F. Spezzano, V. S. Subrahmanian, Asonam 2016 tutorials: Tutorial 1: Evolutionary computation for community detection in complex networks, in: IEEE/ACM International Conference on Advances in Social Networks Analysis δ Mining, 2016.

[20] L. Ma, M. Gong, L. Jie, Q. Cai, L. Jiao, Multi-level learning based memetic algorithm for community detection, Applied Soft Computing Journal 19 (2) (2014) 121–133.

[21] M. Gong, Q. Cai, X. Chen, L. Ma, Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition, IEEE Transactions on Evolutionary Computation 18 (1) (2014) 82–97.

[22] P. M. Zadeh, Z. Kobti, A multi-population cultural algorithm for community detection in social networks, Procedia Computer Science 52 (1) (2015) 342–349.

[23] J. D. Ser, E. Osaba, D. Molina, X. Yang, S. Salcedosanz, D. Camacho, S. Das, P. N. Suganthan, C. A. C. Coello, F. Herrera, Bio-inspired computation: Where we stand and what's next, Swarm and evolutionary computation 48 (2019) 220–250.

[24] C. Pizzuti, Ga-net: A genetic algorithm for community detection in social networks, in: International Conference on Parallel Problem Solving from Nature: Ppsn X, 2008.

[25] C. Pizzuti, A multiobjective genetic algorithm to find communities in complex networks, IEEE Transactions on Evolutionary Computation 16 (3) (2012) 418–430.

[26] A. Clauset, M. E. J. Newman, C. Moore, Finding community structure in very large networks, Physical Review E 70 (6) (2004) 066111.

[27] J. Jin, Fast community detection by score, Annals of Statistics 43 (2) (2015) 672–674.

[28] U. N. Raghavan, R. Albert, S. R. T. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Physical Review E 76 (3) (2007) 036106–036106.

[29] X. Wen, W. N. Chen, L. Ying, T. Gu, J. Zhang, A maximal clique based multiobjective evolutionary algorithm for overlapping community detection, IEEE Transactions on Evolutionary Computation 21 (3) (2017) 363–377.

[30] H. V. Lierde, G. Chen, T. W. S. Chow, Scalable spectral clustering for overlapping community detection in large-scale networks, IEEE Transactions on Knowledge and Data Engineering PP (99) (2019) 1–1.

[31] A. Biswas, B. Biswas, Fuzag: Fuzzy agglomerative community detection by exploring the notion of self-membership, IEEE Transactions on Fuzzy Systems PP (99) (2018) 1–1.

[32] P. Sah, J. Mann, S. Bansal, Disease implications of animal social network structure: A synthesis across social systems, Journal of Animal Ecology 87 (3).

[33] Q. Li, J. Zhong, Q. Li, C. Wang, Z. Cao, A community merger of optimization algorithm to extract overlapping communities in networks, IEEE Access 7 (2019) 3994–4005.

[34] L. Andrea, F. Santo, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Physical Review E Statistical Nonlinear δ Soft Matter Physics 80 (2) (2009) 016118.

[35] A. D. Mccarthy, T. Chen, R. Rudinger, D. W. Matula, Metrics matter in community detection.

[36] A. Arenas, A. Díazguilera, Synchronization and modularity in complex networks, European Physical Journal Special Topics 143 (1) (2007) 19–25.

[37] H. Jin, M. Liang, The hierarchical network topology management system based on managed object and view mechanism, AASRI Procedia 9 (2014) 12–18.

[38] 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: Can geographic isolation explain this unique trait?, Behavioral Ecology δ Sociobiology 54 (4) (2003) 396–405.