

Article

# Large-Scale Complex Network Community Detection Combined with Local Search and Genetic Algorithm

Desheng Lyu <sup>1,2,\*</sup>, Bei Wang <sup>3</sup> and Weizhe Zhang <sup>1,4</sup>

<sup>1</sup> Key Laboratory of Interactive Media Design and Equipment Service Innovation, Ministry of Culture and Tourism in China (Harbin Institute of Technology), Harbin 150001, China; wzzhang@hit.edu.cn

<sup>2</sup> Department of Media Technology and Art, School of Architecture, Harbin Institute of Technology, Harbin 150001, China

<sup>3</sup> Department of Public Foreign Language Teaching and Research, Harbin Normal University, Harbin 150080, China; wangb\_harbin@126.com

<sup>4</sup> Cyberspace Security Research Center, Pengcheng Laboratory, Shenzhen 518055, China

\* Correspondence: deshengl@hit.edu.cn

Received: 27 December 2019; Accepted: 16 April 2020; Published: 30 April 2020



**Abstract:** With the development of network technology and the continuous advancement of society, the combination of various industries and the Internet has produced many large-scale complex networks. A common feature of complex networks is the community structure, which divides the network into clusters with tight internal connections and loose external connections. The community structure reveals the important structure and topological characteristics of the network. The detection of the community structure plays an important role in social network analysis and information recommendation. Therefore, based on the relevant theory of complex networks, this paper introduces several common community detection algorithms, analyzes the principles of particle swarm optimization (PSO) and genetic algorithm and proposes a particle swarm-genetic algorithm based on the hybrid algorithm strategy. According to the test function, the single and the proposed algorithm are tested, respectively. The results show that the algorithm can maintain the good local search performance of the particle swarm optimization algorithm and also utilizes the good global search ability of the genetic algorithm (GA) and has good algorithm performance. Experiments on each community detection algorithm on real network and artificially generated network data sets show that the particle swarm-genetic algorithm has better efficiency in large-scale complex real networks or artificially generated networks.

**Keywords:** large-scale complex network; community structure; community detection algorithm; particle swarm-genetic algorithm

## 1. Introduction

The various characteristic attributes and various rich relational structures contained in complex networks reflect the potential regularity of real systems in the real world. At present, discovering and researching community structures in networks has become a hot topic in complex network research. Community discovery technology mainly studies how to detect and analyze the community structure contained in a large number of complex networks, including discipline theory in the fields of graph theory, data mining, machine learning, statistical analysis and pattern recognition [1]. Discovering and studying the community structure in various types of complex networks is of great value to social life and human practice. Therefore, research on community detection algorithms in large-scale complex networks is an important research topic, both in terms of theoretical significance and practical value.

From the current research status, as the scale and complexity of the network continue to expand, it is becoming more and more difficult to find a satisfactory community structure through

traditional community detection algorithms or a single artificial intelligence optimization algorithm. Although some achievements have been made in the improvement and application research of community detection algorithms, in the existing community detection algorithms actually applied in large-scale complex networks, it is often difficult to meet the actual network requirements. Therefore, this paper focuses on the study of community structure in large-scale complex networks. Based on a comparative study of various existing community detection algorithms, this paper proposes a corresponding community detection algorithm, and combines artificial and real network data to verify and analyze the algorithm. The specific research contents include: (1) Introduce the relevant theories of complex networks, and introduce the two community structure quality evaluation indicators used in this paper: network module degree and normalization mutual information (NMI); (2) focus on the analysis of several common community detection algorithms; (3) Introduce the principle of particle swarm optimization algorithm and genetic algorithm, and introduce the corresponding process; (4) Based on particle swarm optimization and genetic algorithm, a hybrid algorithm strategy is proposed. The test function is used to test the single and hybrid algorithms, and the proposed particle swarm-genetic hybrid algorithm improves the overall performance of the algorithm [2,3]. By testing the community detection algorithms on real networks and artificially generated network data sets, it is verified that the particle swarm-genetic algorithm has certain advantages in the accuracy of community detection compared with other algorithms. The results show that combined with the advantages of particle swarm optimization and genetic algorithm, the proposed particle swarm-genetic hybrid algorithm has certain theoretical significance and application value for community exploration research in large-scale complex networks.

## 2. Prior Research

Research topics on community detection have always been a hot topic for scholars, attracting researchers in many disciplines, including mathematics, sociology, physics, statistics, computer science and biology. In the process of research, new methods or theories about the exploration of complex network community structure are constantly being proposed, and the relevant research results are also fruitful. With the deepening of research, many principles and algorithms for complex network community detection have been developed. In [4], the authors propose an effective overlapping community detection algorithm using the seed expansion method. The algorithmic idea is to find good seeds and then greedily expand these seeds according to community indicators. A new seeding strategy was developed for the personalized PageRank clustering scheme, which optimizes the conductivity community score. In [5], the author introduces two new algorithms, LPAwb+ and DIRTLPAwb+, to maximize the weighting module in the two-party network. LPAwb+ and DIRTLPAwb+ can reliably identify partitions with higher modular scores. In [6], the author proposes a Spark-based parallel incremental dynamic community detection algorithm (PIDCDS), which can maximize the total persistence of vertices in the network to discover the community structure. PIDCDS uses parallel computing on GraphX to calculate the special permanent as a community partitioning indicator. In [7], the author proposes an image segmentation algorithm based on graph-based community detection. The image is represented as a non-directional weighted graph on which community detection is performed, with each community corresponding to a segment in the image. In [8], in order to speed up the optimization process and more effectively discover multiple important community structures, the authors propose a multi-objective evolutionary algorithm that uses problem-specific genetic variation and population crossing and problem-specific initialization to evaluate. In [9], for most existing methods, including some methods to convert the problem into a classic setting of community detection in a simplex network, the author proposes a new method, which includes a seed-centric algorithm adaptation. In [10], the authors propose a novel multi-target bat algorithm that uses the mean shift algorithm to generate an initial population to obtain a high-quality solution. By giving the natural bat speed, frequency, loudness and pulsation rate another meaning, the algorithm's operators are applied to community detection problems in social dynamic networks.

For particle swarm optimization, it has the advantages of fast convergence, high precision and the advantages of genetic algorithm with random global search ability and good generalization ability. Many researchers at home and abroad have applied them in different fields. In [11], the authors used a particle swarm optimization-based approach to train neural networks (NN-PSO) and applied it to building structure prediction. The NN-PSO classifier can solve the problem of predicting failure of multilayer-reinforced concrete building structures by detecting the failure probability of future multilayer RC building structures. In [12], the authors applied PSO to deep brain stimulation (DBS) treatment. The DBSA was programmed by developing a particle swarm optimization algorithm to use a single population of particles representing the electrode configuration and stimulus amplitude. The algorithm provides a computationally efficient way to program DBS systems, especially for systems with higher electrode counts. In [13], in order to solve the location management problem, the author proposes a binary particle swarm optimization algorithm (BPSO) using the optimal reporting cell planning technique, in order to reduce the management costs of position generated during the tracking process of locating users to the cellular network. In [14], the authors propose particle swarm optimization for trajectory optimization. Simulation experiments show that the proposed algorithm successfully optimizes the trajectory while satisfying the constraints and is unlikely to converge to the local minimum. In [15], the redundancy strategy can be selected for a single subsystem. The author applies the genetic algorithm to solve the redundant allocation problem of the series-parallel system. In [16], the author uses genetic algorithm to design a multi-task linear continuous time system robust controller with guaranteed cost. In [17], the authors used genetic algorithms to obtain the best solution to minimize the annual value of the power system during the analysis. In [18], the author applied genetic algorithm and particle swarm optimization algorithm to the selection of the optimal parameter combination of laser micro-marking. By using genetic algorithm and particle swarm optimization, the optimal setting of input constraints was predicted.

From the application research results of particle swarm optimization and genetic algorithm, the particle swarm optimization algorithm has good local search performance, and the genetic algorithm has good global search ability. Therefore, this paper combines the two, and the proposed hybrid algorithm improves the performance of the algorithm based on the performance of particle swarm and genetic algorithm. Compared with previous research, the hybrid algorithm proposed in this paper has better accuracy in community detection. And with the increase of network scale, the research method in this paper is also significantly better than the traditional method in terms of calculation speed.

### 3. Proposed Method

#### 3.1. Complex Network Community Detection Concept

##### (1) Community structure of complex networks

In addition to the characteristics of preference connection, small world effect, power law distribution and scale-free topology, complex networks also exhibit community structure characteristics. In order to better study and analyze the community structure of the network, the community is further divided into two types: strong community and weak community. For a community  $C = \{v_1, v_2 \dots v_c\}$ , any node in the community satisfies:

$$k_i^{in} > k_i^{out} \quad \forall v_i \in C \quad (1)$$

where  $k_i^{in}$  is the number of connections between the node  $v_i$  and the internal nodes of the community and  $k_i^{out}$  is the number of connections between the node  $v_i$  and the external nodes of the community, apparently  $k_i = k_i^{in} + k_i^{out}$ . In this way, community  $C$  is called a strong community. If the community meets:

$$\sum_i^c k_i^{in} > \sum_i^c k_i^{out} \quad \forall v_i \in C \tag{2}$$

Then community  $C$  is called a weak community.

(2) Community structure quality evaluation indicators

i. Modular

The network module degree is defined as follows: the difference between the number of connected edges in the network module and the number of connected edges in the random case refers to the proportion of the number of connected sides of the entire network. Assume that for Network  $C$ , if it has a clear modular structure, the network is a random network of the same situation. Then, the sum of the number of internal sides of each module in the network  $C$  must be greater than the sum of the number of connected sides in each community in the network. Based on the above assumptions, the following expression can be defined:

$$Q = \frac{1}{2m} \sum_{vw} \left[ A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) = \sum_i (e_{ii} - a_i^2) \tag{3}$$

$$e_{ii} = \frac{1}{2m} \sum_{vw} A_{vw} \delta(c_v, i) \delta(c_w, i) \tag{4}$$

$$a_i = \frac{1}{2m} \sum_v k_v \delta(c_v, i) \tag{5}$$

where  $A$  represents the adjacency matrix of network  $C$ . If there is an edge between node  $v$  and node  $w$ , then  $A_{v,w} = 1$ , otherwise 0.  $m$  represents the total number of connected sides of the network, and  $2m$  represents the total number of degrees of the network.  $k_v, k_w$  represent the degrees of nodes  $v, w$ , respectively.  $e_{ii}$  represents the ratio of all edges in community  $i$  to all sides of the entire network.  $a_i$  represents the number of connections for the  $i$ -th community and all other nodes within the community. The  $\delta$  function indicates that when the nodes  $v, w$  belong to the same community, it is 1, otherwise it is 0, which ensures that only the connected edges in the community are considered.

ii. NMI

Mutual information is a measure of the correlation between random variables in probability theory and information theory and is usually used to measure the degree of dependence between each other. It is introduced into the community structure quality evaluation and normalized, which is normalization mutual information [19]. In the actual calculation, a corresponding hybrid matrix  $N$  is generated according to the community division of the network and the community partition obtained by the community detection algorithm, wherein the element  $N_{ij}$  in the matrix  $N$  represents the shared node between the actual community  $C_i$  and the community  $C_j$  obtained by the algorithm. The specific formula is as follows:

$$NMI(A;B) = \frac{I(A;B)}{\sqrt{H(A)H(B)}} = \frac{-2 \sum_{i=1}^{|C(A)|} \sum_{j=1}^{|C(B)|} \log\left(\frac{n \cdot N_{ij}}{N_i N_j}\right)}{\sum_{i=1}^{|C(A)|} N_i \log\left(\frac{N_i}{n}\right) + \sum_{j=1}^{|C(B)|} N_j \log\left(\frac{N_j}{n}\right)} \tag{6}$$

$A$  and  $B$ , respectively represent the real division of the network and the division obtained by the community detection algorithm.  $|C(A)|$  and  $|C(B)|$ , respectively correspond to the number of communities in the actual community number of the network and the community detection algorithm and  $N_i$  represents the matrix. The sum of the elements of the  $i$ -th row of  $N$  and  $N_j$  represents the sum of the elements of the  $j$ -th column in the matrix  $N$ .  $NMI$  has a good ability to recognize the network

of known actual communities. The larger the *NMI* value, the better the community partitioning effect obtained by the community detection algorithm. When the *NMI* value is 1, it indicates that the discovered community is completely consistent with the real community.

### (3) Community detection algorithm

#### i. GN algorithm

The GN algorithm is a cohesive community structure discovery algorithm. Based on the characteristics of high cohesion within communities and low cohesion among communities, the algorithm gradually removes the edges between communities to obtain a relatively cohesive community structure. The algorithm uses the concept of edge median to detect the position of an edge. The edge median of an edge is defined as the number of times the shortest path between all vertices on the network passes through the edge. The basic idea is to realize the division of the network by continuously deleting the edge with the largest number of edges in the network until all the edges in the network are deleted. It is based on the internal links of the community, and the shortest path through them is relatively small; while the links between the communities have relatively shortest paths. The flow of the GN algorithm is as follows:

- (a) Calculate the edge number of each edge in the network.
- (b) Delete the edge with the largest median.
- (c) Recalculate the number of edges of the remaining edges in the network.
- (d) Repeat steps (b) and (c) until the nodes in the network are divided into separate communities.

#### ii. Fastgreedy algorithm

The GN algorithm is only suitable for small networks because it needs to repeatedly calculate the edge number of the edge and has a large time overhead. Compared to the Internet, e-mail networks and other large networks with hundreds of millions of nodes, it is clear that the GN algorithm cannot meet the requirements. Based on this, the Fastgreedy algorithm is a fast search algorithm developed on the basis of the GN algorithm [20,21]. In fact, the essence of the algorithm is one of the cohesive algorithms based on greedy thought. The steps of the Fastgreedy algorithm are as follows:

- (a) Initialize the network into  $n$  communities and consider each node as an independent community.
- (b) Calculate the module  $Q$  value corresponding to the combination of the two communities, and increase or decrease the maximum or the maximum along the module increment  $\Delta Q$  community consolidation in a small direction:

$$\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j) \quad (7)$$

- (c) Repeat (b) until all the communities in the network are merged into one large community, the algorithm stops, and the community partition result corresponding to the maximum module degree in the merge process is found.

#### iii. LFM algorithm

The LFM algorithm is different from the community discovery algorithm that understands the community structure from a global perspective. The basic premise of the algorithm is that the community structure is a local structure, that is, the community relationship to which the node belongs is most related to its neighbor nodes. This idea is consistent with the current situation of many large-scale networks. For example, in social networks, individual community relationships do not need to determine their community affiliation based on understanding the entire human relationship network. The LFM algorithm defines a fitness function:

$$f_G = \frac{k_{in}^G}{(k_{in}^G + k_{out}^G)^\partial} \tag{8}$$

where  $k_{in}^G$  and  $k_{out}^G$  represent the degree in the community  $G$  (the sum of the weights of the edges in the community  $G$ ) and the degrees outside the community (the sum of the weights between the  $G$  and other communities), and  $\partial$  is a positive resolution parameter. Used to control the size of the community, so the LFM algorithm is also a community discovery algorithm that can detect hierarchical community structures.

The fitness change before and after node  $A$  joins community  $G$  is:

$$f_G^A = f_{G+(A)} - f_{G-(A)} \tag{9}$$

The LFM algorithm defines the natural community structure of the node: for a sub-graph  $G$  where the node  $A$  is located, if a node is added or a node is deleted, the fitness is reduced and the sub-graph structure  $G$  is called a natural community.

### 3.2. Particle Swarm Optimization

#### (1) Principles of particle swarm algorithm

Particle swarm optimization is essentially a random search optimization algorithm based on iterative computation [22]. It belongs to a relatively novel optimization method. It has some similar calculation methods with genetic algorithms. The difference is that it does not contain basic operations such as crossover and mutation. Instead, it searches for the optimal solution of the predation behavior of the flock. Each time the particle swarm optimization algorithm performs an iterative optimization, all the particles will be dynamically adjusted according to the superior value of the contemporary individual and the excellent value of the group to update their speed and position. The main formula is as follows:

$$V_{id}^{k+1} = \omega V_{id}^k + c_1 r_1 (P_{id}^k - X_{id}^k) + c_2 r_2 (P_{gd}^k - X_{gd}^k) \tag{10}$$

$$X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1} \tag{11}$$

where  $V_{id}^k$  represents the velocity value of the  $d$ -th dimension of the particle  $i$  in the  $k$ -th iteration update;  $X_{id}^k$  represents the position coordinate of the  $d$ -th dimension of the particle  $t$  in the  $k$ -th iteration update;  $P_{id}^k$  represents the particle  $i$  in the  $k$ -th iteration update. The position coordinate of the individual extreme point of the  $d$ -th dimension;  $P_{gd}^k$  represents the position coordinate of the global extreme point of the  $d$ -th dimension of the particle  $i$  in the  $k$ -th iteration update;  $r_1$  and  $r_2$  are randomly selected on  $[0,1]$ . The number is used to adjust the self-cognitive experience part of the particle;  $c_1, c_2$  are learning factors used to adjust the social cognitive part of the particle.

#### (2) Particle swarm algorithm flow

The specific process of the particle swarm algorithm is as follows:

- (1) Initialization: Set the number of iterations of the algorithm, the learning factor and the initial velocity of all particles randomly generated and the position in the solution space.
- (2) Calculate the fitness value, individual excellent value and group excellent value of the particles in the population.
- (3) Update the velocity and position of the particles in the population, fitness function values, individual extremum and group extremum.

- (4) Check the overall performance of the new particle swarm. If the optimization effect is not achieved, skip to step (2) and continue.
- (5) Achieve the set number of iterations, end and output the optimal solution in the particle swarm (the optimal solution is the individual value with the best fitness value).

### 3.3. Genetic Algorithm

#### (1) Principles of genetic algorithms

The genetic algorithm is based on the selection of the natural environment and the reproductive, crossover and genetic mutations that occur in the genetic process, simulating the evolutionary process of natural organisms and finding the optimal solution [23]. The genetic algorithm directly encodes the structural object, does not need to derive the index function, and defines the continuity of the fitness function. It can process multiple solutions in the search space together and has a high global search capability. The probabilistic guidance optimization method can autonomously determine the search space and search direction, independent of the determined rules; for those nonlinear, large-scale, no specific analytical or multiple index function optimization problems, compared to other types. The optimization method, the algorithm can get better optimization results.

#### (2) Genetic algorithm flow

The performance of the genetic algorithm depends largely on the selection of the algorithm control parameters, and the termination condition of the algorithm generally reaches the maximum number of iterations or reaches the specified error range. The specific process of the genetic algorithm is as follows:

- (1) Coding: The coding of genetic algorithm is the genetic representation of the solution. It is usually expressed as a numeric string or a string. The solution data  $x$  in the solution space is the expression of the genetic algorithm. The binary coding method is the most important kind of coding method in the genetic algorithm.
- (2) Initialization: randomly generate  $N$  initial binary strings, that is, a total of  $N$  individuals in the initial population, and perform initialization operations on these string structures. The specific problem of the length of the string is specifically analyzed, and the above-mentioned individuals are the initial groups  $P(t)$ , form a set of candidate solutions.
- (3) Individual evaluation: Calculate the fitness of each individual in the group  $P(t)$ , and the individuals are sorted according to the size of the fitness value.
- (4) Selection operation: The selection operator is used for the parent group. The method is to directly inspect the selected high-quality individuals to the next generation for recombination and mutation. This also reveals the principle of survival of the fittest in the process of biologic evolution.
- (5) Crossover operation: also known as recombination, the crossover operator is applied to the group, and two pairs of parent individuals are randomly selected from the population according to a certain crossover probability, and part of the structural data are exchanged to form two new individuals.
- (6) Mutation operation: The mutation operator is used for the population, and some gene values of the individual are changed according to the small mutation probability, and the mutation operation can maintain the genetic diversity.
- (7) Iterative termination judgment: If the number of iterations  $t < \text{MAXGEN}$ , then let  $t = t + 1$  and go back to step (2); if the iteration number  $t = \text{MAXGEN}$ , directly output the maximum fitness value, corresponding to its value. The individual is the optimal individual, the optimal solution is output, and the algorithm terminates.

### 3.4. Particle Swarm-Genetic Hybrid Algorithm

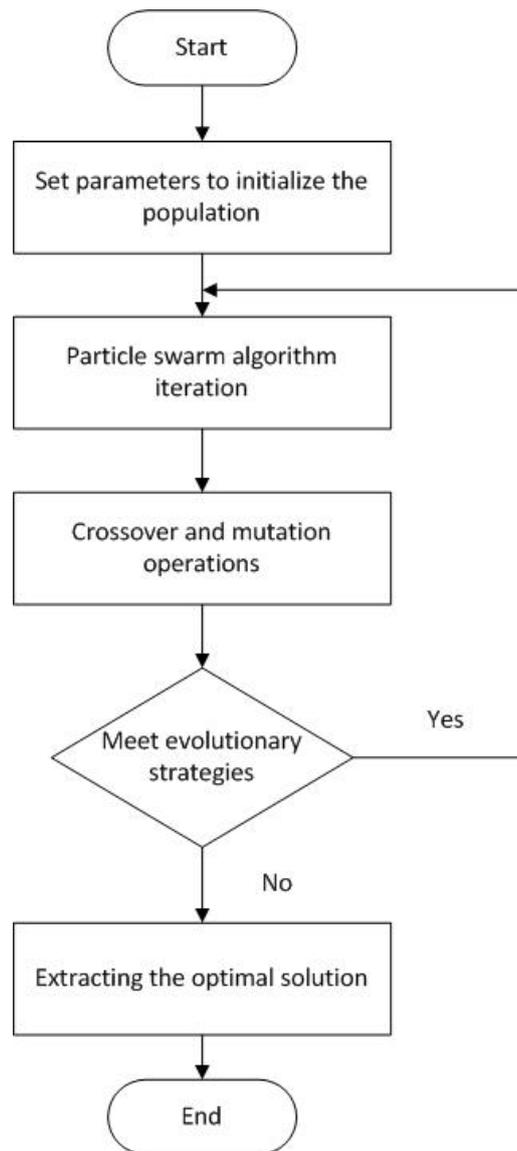
#### (1) Algorithm hybrid strategy

The mixing of algorithms needs to follow some combination principles so the advantages of the original evolutionary algorithms can be effectively utilized, the convergence speed can be accelerated and the quality of the optimal feasible solution solved by the hybrid algorithm is higher than that of the original algorithm—thus achieving the purpose of improving the algorithm. The combination of particle swarm optimization and genetic hybrid algorithm designed in this paper is designed to share the historical optimal feasible solution searched by each particle in the particle swarm optimization algorithm with the genetic algorithm. The genetic operator searches and evolves based on these feasible solutions. The genetic operator updates these historical optimal feasible solutions according to the specified evolutionary principle, which can not only maintain the good local search performance of the particle swarm algorithm, but also take advantage of the good global search ability of the genetic algorithm.

#### (2) Hybrid algorithm flow

The particle swarm-genetic hybrid algorithm (PSO-GA) flow proposed in this paper is shown in Figure 1. The main steps are as follows:

- (1) Set the parameters of the particle swarm and genetic hybrid algorithm. These parameters mainly include the population size POPSIZE, the iteration number Tmax of the algorithm operation, the self-adjustment range of the adaptive inertia weight factor  $W$ , and the variation range of the asynchronous change learning factors C1 and C2.
- (2) Initialize the population: Generate POPSIZE particles according to the parameters set in step 1). The particle swarm algorithm, the genetic algorithm's operation object and the evolution strategy are not the same. Two initial chromosome codes should be copied for subsequent use. Evolutionary calculations that update the current fitness values of the particles.
- (3) Evolutionary strategy: Due to the particularity of the genetic algorithm evolution preservation strategy designed in this paper, both the crossover and the mutation operator will only evolve to better individuals. Therefore, the operation object of the genetic algorithm is the historically optimal individual chromosome code that each particle has found in the iterative search of the particle swarm. The particle swarm algorithm operates another chromosome code. When the corresponding fitness of this chromosome code is higher, the chromosome code covering the operation of the genetic algorithm is updated.
- (4) Perform particle swarm algorithm iteration: update the position vector and velocity vector of the particle. If the fitness corresponding to the new particle is higher than the local historical optimal feasible solution or the global historical optimal feasible solution, then the replacement is performed.
- (5) Crossover operation of genetic algorithm: Using a random matching strategy, the matched two particle individuals cross-evolve the specified chromosomal genes and compare the evolution results with the old individuals to decide whether to replace or discard. If a better individual is evolved, the local best is compared to the current global best individual, and if the individual is better than the current global best feasible solution, the global best feasible solution is replaced, otherwise no action is taken.
- (6) Variation operation of genetic algorithm: Similar to the strategy of step (5), if the fitness corresponding to the new particle is higher than the local historical optimal feasible solution or the global historical optimal feasible solution, then the corresponding replacement is performed.
- (7) Judgment: If the evolution stop condition is satisfied, the execution of the algorithm is stopped, otherwise it is returned to step (4) to repeat the execution.
- (8) Extraction: After the evolution stops, the best feasible solution of the original population is the best feasible solution to the target calculation problem.



**Figure 1.** Flow chart of particle swarm optimization-genetic algorithm (PSO-GA) hybrid algorithm.

## 4. Experiments

### 4.1. Experimental Data Set

#### (1) Real network data set

This article uses five common real-world network data sets: the Karate Network, the Dolphin Network, the Book Network, the Football Network and the Scientist Partnership Network. The specific information is shown in Table 1.

**Table 1.** Real network data set.

Network Name	Number of Nodes	Number of Sides	Network Description
Karate	34	78	Karate club
Dolphin	62	159	Dolphin social network
Polbook	105	441	American Political Book Network
Football	115	613	American college football game
Netscience	1589	2742	Scientist partnership network

(2) Manually generate network data sets

In this paper, the GN benchmark network data set is used to test the PSO-GA hybrid algorithm and the common community detection algorithm, and an accuracy metric based on information theory is adopted as the evaluation criterion of the clustering result. The GN reference artificial network can be defined as GN (C, s, d, Zout), where C represents the number of communities in the network, s represents the number of nodes in each community, and d represents each node degree in the network. Zout represents the number of links between each node and the nodes outside the community.

4.2. Test Function

In order to verify the feasibility, convergence and stability of the PSO-GA hybrid algorithm proposed in this paper, the following unconstrained test functions are used to test a single particle swarm optimization algorithm and genetic algorithm and the algorithm.

$$F(x_1, x_2) = (x_1^2 + x_2^2)^{\frac{1}{4}} \left( \sin^2 \left( 50 \left( (x_1^2 + x_2^2)^{\frac{1}{10}} \right) + 1 \right) \right) \tag{12}$$

Among them,  $x_1, x_2 \in [-100, 100]$ , the minimum value is 0.

4.3. Parameter Settings

The experimental parameters are set as follows:

The population size is 100, the maximum evolution number of hybrid and genetic algorithms is 100, and the maximum number of iterations of a single particle swarm algorithm is 1000. The lower limit of the adaptive dynamic inertia weight is 0.4, and the upper limit is 0.9. The initial value of the learning factor C1 is 2.5 and the cutoff value is 0.5; the initial value of the learning factor C2 is 0.5 and the cutoff value is 2.5.

5. Results and Discussions

5.1. Comparison of Single Algorithm and Hybrid Algorithm

By performing 10 experimental statistics on the test function, the obtained comparison results are shown in Tables 2 and 3. Among them, the “optimal average (particle swarm algorithm)” is the average value of the optimal values searched for 10 experimental tests after 1000 iterations; “optimal average (genetic algorithm)” and “optimal average (mixed algorithm)” statistic is the average value of the optimal values searched for 10 experimental tests after 100 generations of evolution.

Table 2. Test results of particle swarm optimization (PSO) algorithm and hybrid algorithm.

	Optimal Average (PSO)	Optimal Average (PSO-GA)
Time(s)	0.109	0.031

Table 3. Test results of genetic algorithm and hybrid algorithm.

	Optimal Average (GA)	Optimal Average (PSO-GA)
Cycles	78	15

It can be seen from Table 2 that the average time required for the hybrid algorithm to search for a feasible solution with specified precision is better than that of a single particle swarm optimization algorithm. This is mainly due to the global search ability of the genetic algorithm in the hybrid algorithm, completing 100 generation evolution. The feasible solution precision found by the hybrid algorithm is basically the same as the single particle swarm algorithm after 1000 iterations. This is because the local search of the hybrid algorithm is mainly done by the particle swarm algorithm.

It can be seen from Table 3 that the hybrid algorithm searches for a feasible solution with a specified precision and requires fewer evolution than a single genetic algorithm. After 100 evolution, the hybrid algorithm finds a higher resolution accuracy than a single genetic algorithm, indicating that the hybrid algorithm’s local search ability is better than a single genetic algorithm. The results show that the combination of particle swarm optimization and genetic algorithm does improve the overall performance of the algorithm, avoids the disadvantages of weak local search ability of a single genetic algorithm, and the global search ability is better than the single particle swarm optimization algorithm, achieving the purpose of algorithm combination.

### 5.2. Comparison of Various Algorithms on the Real Network

In order to test the maximum modularity of different algorithms, this paper applies PSO-GA hybrid algorithm, GN algorithm, Fastgreedy algorithm, LFM algorithm to five real network data sets of karate network, dolphin network, book network, football network and scientist cooperative network. The algorithm was tested. The experimental results obtained are shown in Figure 2. Among them, the ordinate Qmax is the maximum modularity of the network. For the calculation formula (3), see the third section. The abscissa is each data set.

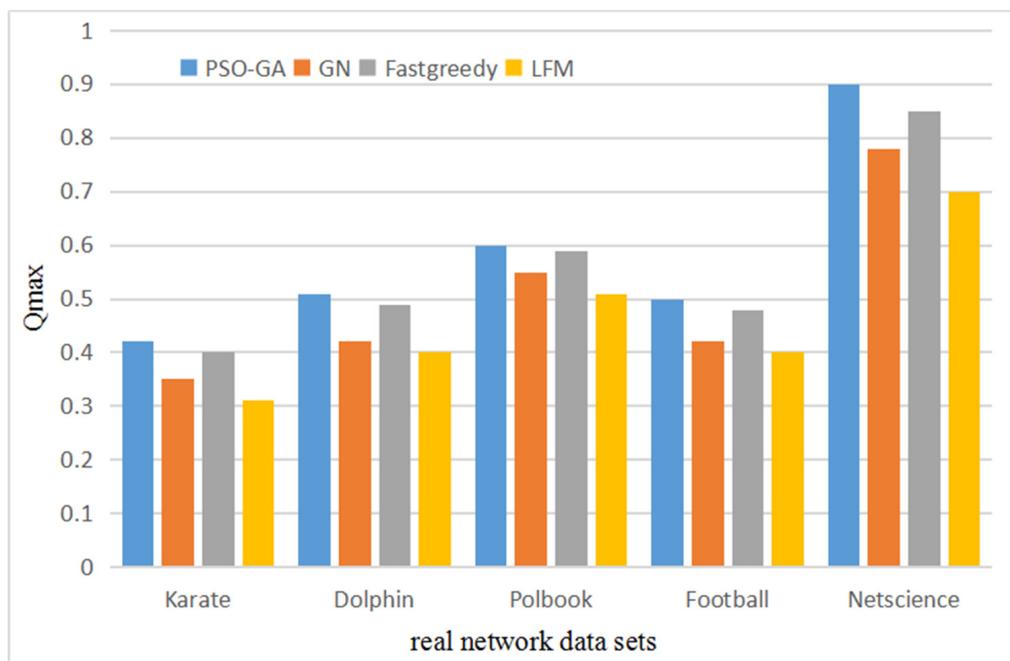


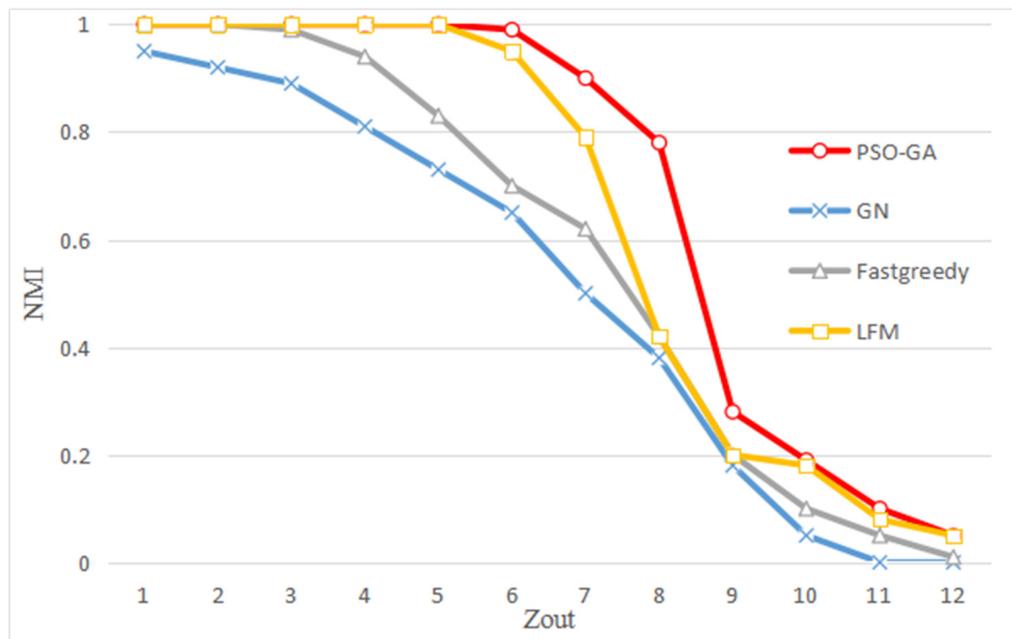
Figure 2. Comparison of results of various algorithms on the real network.

It can be seen from Figure 2 that the PSO-GA hybrid algorithm is more modular than the modularity-based community detection algorithm GN and Fastgreedy, indicating that the hybrid algorithm is effective in community detection. Because the algorithm is too random, the LFM algorithm can’t get a good result in many cases. The community partitioning results obtained by the algorithm vary greatly each time, so it can’t effectively find the community partitioning mode with higher module degree. Due to the huge time consumption of the GA algorithm itself, it is not suitable for large-scale complex networks and the Fastgreedy algorithm based on the GA algorithm has higher module value on most networks. The results show that the PSO-GA algorithm proposed in this paper has certain advantages in the accuracy of community detection compared with other algorithms.

### 5.3. Comparison of Various Algorithms on the Artificial Generation Network

In order to test the clustering accuracy of different algorithms, this paper uses the widely used random network GN (4, 32, 16, Zout). It can be seen that as Zout increases, the network community

structure becomes more and more blurred, and it also brings more and more challenges to community detection algorithms. When  $Z_{out} > 8$ , the community structure of the network is considered to be very vague. The experiment uses PSO-GA hybrid algorithm, GN algorithm, Fastgreedy algorithm and LFM algorithm to cluster 50 random networks, respectively, and the average accuracy obtained is shown in Figure 3.



**Figure 3.** Comparison of results of various algorithms on the artificially generated network.

As can be seen from Figure 3, the PSO-GA algorithm has the highest NMI accuracy on the GN reference network, the LFM algorithm is slightly lower, and the Fastgreedy and GN algorithms are much lower. The LFM algorithm considers the idea that the community structure is a local structure conforms to many large-scale networks. Therefore, its NMI accuracy is higher on the GN reference network, while the module-based community detection algorithms GN and Fastgreedy are relatively low. The results show that the clustering accuracy of the PSO-GA algorithm proposed in this paper is better than the three algorithms.

Computing speed is another important index to evaluate the performance of community detection algorithms. This paper uses a random network GN ( $C, 100, 16, 5$ ), where the number of nodes  $s$  in each community is 100, the degree  $d$  of each node in the network is 16, and the number of links between each node and nodes outside the communities  $Z_{out}$  is 5. As the number of communities  $C$  in the network increases, the PSO-GA hybrid algorithm is tested, and the experimental results obtained are shown in Figure 4.

It can be seen from Figure 4 that as the network scale increases, the running time of the PSO-GA algorithm increases. When the size of the word community is constant, the running time of the algorithm is approximately proportional to the network size. The size of the community in a large-scale real network is generally much smaller than the entire network scale. Therefore, for a large-scale complex real network, the efficiency of the PSO-GA algorithm is very high. The results show that the method in this paper has good feasibility in large-scale community networks.

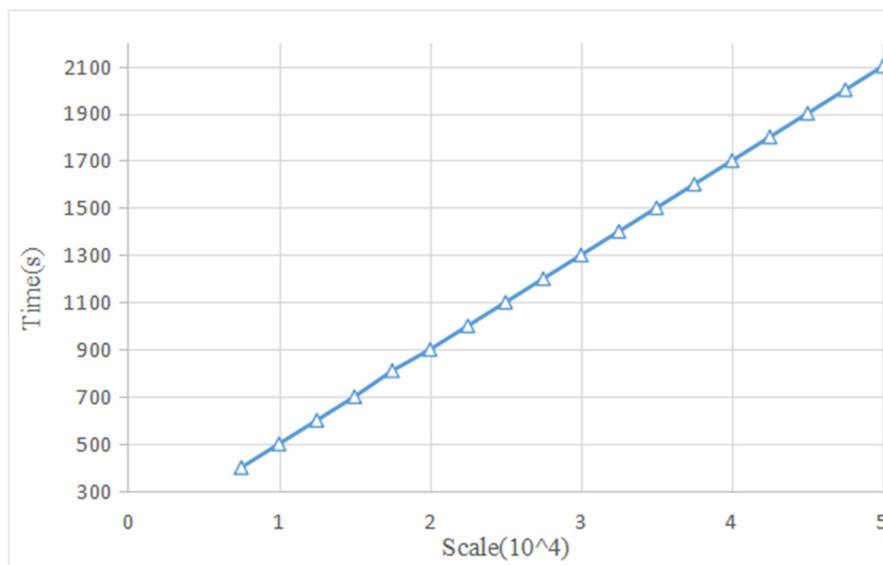


Figure 4. PSO-GA algorithm runtime test on a manually generated network.

## 6. Conclusions

In recent years—with the popularity of online social networks and the popularity of smart mobile devices—more and more people regard online social networking as an important lifestyle, and online social networks have become massive. As an important direction of social network research, community detection is of great significance in studying the characteristics of network structure, analyzing user relations, exploring the way of message dissemination and mastering the trend of public opinion. Therefore, this thesis deeply studies the community detection algorithm of large-scale complex networks and proposes a particle swarm-genetic hybrid algorithm.

The main work of this paper is as follows:

- (1) Introduce the related concepts of complex network community detection and introduce the two-community structure quality evaluation indicators used in this paper: network module degree and NMI. Three common community detection algorithms GN, Fastgreedy and LFM are analyzed, and the corresponding principles and processes are introduced in the paper.
- (2) Analyze and introduce the principle and flow of particle swarm algorithm and genetic algorithm. A hybrid algorithm strategy combining the two is proposed, and the basic flow of the particle swarm-genetic hybrid algorithm is developed according to the strategy.
- (3) The single and hybrid algorithms are tested by test function, and the effectiveness of the proposed algorithm is verified. The algorithm and three other common community detection algorithms are tested on real networks and artificially generated network datasets. The results show that the combination of particle swarm optimization and genetic algorithm is superior to other algorithms and does improve the overall performance of the algorithm, avoids the disadvantages of weak local search ability of a single genetic algorithm, and the global search ability is better than the single particle swarm optimization algorithm.

**Author Contributions:** Writing—Original draft: D.L.; Writing—review & editing: B.W.; Data curation and formal analysis: W.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by MOE (Ministry of Education in China) Project of Humanities and Social Sciences (No. 17YJAZH058), the Fundamental Research Funds for the Central Universities (No. HIT.HSS.201844) and the MOCT (Ministry of Culture and Tourism in China) Funded Project for Key Laboratory.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

PSO	particle swarm optimization
GA	genetic algorithm
NMI	normalization mutual information
PIDCDS	parallel incremental dynamic community detection algorithm
NN-PSO	particle swarm optimization-based approach to train neural networks
DBS	deep brain stimulation
BPSO	binary particle swarm optimization algorithm
PSO-GA	particle swarm-genetic hybrid algorithm

## References

1. Tan, H.; Wu, Y.; Zhang, H. Research on community discovery method for complex authorized networks. *J. Chin. Inf. Process.* **2018**, *32*, 111–119.
2. Belhocine, A.; Omar, W.Z.W. A numerical parametric study of mechanical behavior of dry contacts slipping on the disc-pads interface. *Int. J. Comput. Appl.* **2017**, *40*, 42–60.
3. Xiong, Z.; Wu, Y.; Ye, C.; Zhang, X.; Xu, F. Color image chaos encryption algorithm combining CRC and nine palace map. *Multimed. Tools Appl.* **2019**, *22*, 31035–31055. [[CrossRef](#)]
4. Whang, J.J.; Gleich, D.F.; Dhillon, I.S. Overlapping community detection using neighborhood-inflated seed expansion. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1272–1284. [[CrossRef](#)]
5. Beckett, S.J. Improved community detection in weighted bipartite networks. *R. Soc. Open Sci.* **2016**, *3*, 140536. [[CrossRef](#)]
6. Wu, B.; Xiao, Y.; Zhang, Y. Parallel incremental dynamic community detection algorithm based on spark. *Qinghua Daxue Xuebao/J. Tsinghua Univ.* **2017**, *57*, 1030–1037.
7. Belim, S.V.; Larionov, S.B. An algorithm of image segmentation based on community detection in graphs. *Comput. Opt.* **2017**, *40*, 904–910. [[CrossRef](#)]
8. Žalik, K.R.; Žalik, B. Multi-objective evolutionary algorithm using problem-specific genetic operators for community detection in networks. *Neural Comput. Appl.* **2018**, *30*, 2907–2920. [[CrossRef](#)]
9. Hmimida, M.; Kanawati, R. Community detection in multiplex networks: A seed-centric approach. *NHM* **2015**, *10*, 71–85. [[CrossRef](#)]
10. Messaoudi, I.; Kamel, N. A multi-objective bat algorithm for community detection on dynamic social networks. *Appl. Intell.* **2019**, *49*, 2119–2136. [[CrossRef](#)]
11. Hore, S.; Hore, S.; Hore, S.; Dey, N.; Ashour, A.S.; Balas, V.E. Particle swarm optimization trained neural network for structural failure prediction of multistoried rc buildings. *Neural Comput. Appl.* **2017**, *28*, 2005–2016.
12. Peña, E.; Zhang, S.; Deyo, S.; Xiao, Y.Z.; Johnson, M.D. Particle swarm optimization for programming deep brain stimulation arrays. *J. Neural Eng.* **2017**, *14*, 016014. [[CrossRef](#)] [[PubMed](#)]
13. Parija, S.R.; Sahu, P.K.; Singh, S.S. Cost reduction in location management using reporting cell planning and particle swarm optimization. *Wirel. Pers. Commun.* **2017**, *96*, 1–21. [[CrossRef](#)]
14. Kim, J.J.; Lee, J.J. Trajectory optimization with particle swarm optimization for manipulator motion planning. *IEEE Trans. Ind. Inform.* **2015**, *11*, 620–631. [[CrossRef](#)]
15. Tavakkolmoghadam, R.; Safari, J.; Sassani, F. Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. *Reliab. Eng. Syst. Saf.* **2017**, *93*, 550–556. [[CrossRef](#)]
16. Sekaj, I.; Veselý, V. Robust output feedback controller design: Genetic algorithm approach. *Ima J. Math. Control Inf.* **2018**, *22*, 257–265. [[CrossRef](#)]
17. Volkanovski, A.; Mavko, B.; Bosevski, T.; Causevski, A.; Cepin, M. Genetic algorithm optimisation of the maintenance scheduling of generating units in a power system. *Reliab. Eng. Syst. Saf.* **2017**, *93*, 779–789. [[CrossRef](#)]
18. Kalita, M.K.; Shivakoti, I.; Ghadai, R.K. Optimizing process parameters for laser beam micro-marking using a genetic algorithm and particle swarm optimization. *Mater. Manuf. Process.* **2017**, *32*, 1–8. [[CrossRef](#)]
19. Ball, K.R.; Grant, C.; Mundy, W.R.; Shafer, T.J. A multivariate extension of mutual information for growing neural networks. *Neural Netw.* **2017**, *95*, 29–43. [[CrossRef](#)]

20. Yang, A.; Li, Y.; Kong, F.; Wang, G.; Chen, E. Security Control Redundancy Allocation Technology and Security Keys Based on Internet of Things. *IEEE Access*. **2018**, *6*, 50187–50196. [[CrossRef](#)]
21. Yang, Y.; Zhong, M.; Yao, H.; Yu, F.; Fu, X.; Postolache, O. Internet of Things for Smart Ports: Technologies and Challenges. *IEEE Instrum. Meas. Mag.* **2018**, *21*, 34–43. [[CrossRef](#)]
22. Du, K.L.; Swamy, M.N.S. Particle Swarm Optimization. In *Search and Optimization by Metaheuristics*; Birkhäuser: Cham, Switzerland, 2016; pp. 153–173.
23. Ali, M.; Zahra, S.T.; Jalal, K.; Saddiqa, A.; Hayat, M.F. Design of Optimal Linear Quadratic Gaussian (LQG) Controller for Load Frequency Control (LFC) using Genetic Algorithm (GA) in Power System. *Int. J. Eng. Work.* **2018**, *5*, 40–49.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).