

Article

MOPIO: A Multi-Objective Pigeon-Inspired Optimization Algorithm for Community Detection

Junliang Shang ¹, Yiting Li ¹, Yan Sun ^{1,*}, Feng Li ¹, Yuanyuan Zhang ² and Jin-Xing Liu ¹

¹ School of Computer Science, Qufu Normal University, Rizhao 276826, China; jlshang@qfnu.edu.cn (J.S.); ssyuan@qfnu.edu.cn (Y.L.); LiFeng28@qfnu.edu.cn (F.L.); sdcavell@qfnu.edu.cn (J.-X.L.)

² School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266000, China; zhangyuanyuan@qut.edu.cn

* Correspondence: sunyan225@qfnu.edu.cn

Abstract: Community detection is a hot research direction of network science, which is of great importance to complex system analysis. Therefore, many community detection methods have been developed. Among them, evolutionary computation based ones with a single-objective function are promising in either benchmark or real data sets. However, they also encounter resolution limit problem in several scenarios. In this paper, a Multi-Objective Pigeon-Inspired Optimization (MOPIO) method is proposed for community detection with Negative Ratio Association (NRA) and Ratio Cut (RC) as its objective functions. In MOPIO, the genetic operator is used to redefine the representation and updating of pigeons. In each iteration, NRA and RC are calculated for each pigeon, and Pareto sorting scheme is utilized to judge non-dominated solutions for later crossover. A crossover strategy based on global and personal bests is designed, in which a compensation coefficient is developed to stably complete the work transition between the map and compass operator, and the landmark operator. When termination criteria were met, a leader selection strategy is employed to determine the final result from the optimal solution set. Comparison experiments of MOPIO, with MOPSO, MOGA-Net, Meme-Net and FN, are performed on real-world networks, and results indicate that MOPIO has better performance in terms of Normalized Mutual information and Adjusted Rand Index.

Keywords: community detection; Pigeon-Inspired Optimization; Pareto-optimal sorting; multi-objective optimization; complex network



Citation: Shang, J.; Li, Y.; Sun, Y.; Li, F.; Zhang, Y.; Liu, J.-X. MOPIO: A Multi-Objective Pigeon-Inspired Optimization Algorithm for Community Detection. *Symmetry* **2021**, *13*, 49. <https://doi.org/10.3390/sym13010049>

Received: 10 December 2020

Accepted: 28 December 2020

Published: 30 December 2020

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Complex systems are common in nature and human society, most of which can be modelled and analyzed by complex networks, such as power network, transport system, epistatic interactions [1], cyber risk assessment model [2], social network, and other areas [3]. In these networks, vertices and edges, respectively, represent elementary units composing complex systems and interactions between units. Therefore, researching properties of complex networks is of great importance for understanding complex systems. With the in-depth research of complex networks, a growing number of properties have been captured [4]. Among them, community structure is the important and famous one [5], which indicates the trend of aggregation of nodes in the network: connections between nodes of the same community are closer, and of the different communities are more sparse [6]. The detection of communities can help to find the functional structure of complex networks, leading to better understanding the corresponding complex system, and hence becomes the hot topic in the field of network science.

In fact, many community detection methods have been proposed so far. Among them, one famous category is based on evolutionary computation, which belongs to artificial intelligent optimization metaheuristics inspired by principles from biology, ethology and so on [3–7]. Evolutionary computation methods are promising in solving complex problems since a simple and efficient evolutionary computation method can be easily developed by

determining the representation for one complex problem, the function to optimize, and the evolutionary strategies of individuals. Compared to classical metaheuristics methods, main advantages of them are that the state space of feasible solutions is exploited fully and the number of communities is automatically determined during the search process.

However, many evolutionary computation methods to solve community detection merely consider single-objective function, which may encounter resolution limit problem and bias toward a given community structure [8–10]. For instance, Zhang et al. [9] proposed the MPSSA algorithm based on particle swarm optimization (PSO) to detect the community structure of complex networks. It introduces both global and tabu local search strategies in order to overcome the resolution limit problem. Gong et al. [8] proposed Meme-Net method which optimizes single modularity density function and combines genetic algorithm (GA) with a hill-climbing strategy as its local search strategy. Meme-Net performs better than classical GAs on community detection, but one limitation that exists within it is its dependence on parameter tuning. In addition, Guo et al. [10] proposed a GA based method LSSGA, which introduces a novel generation strategy for initial population. LSSGA also uses an effective mutation operator according to label propagation and local structure similarity to keep a balance between diversity and convergence. Understandably, multiple objective optimizations tend to evaluate community structure from different perspectives [6,11–19]. Pizzuti et al. [11] presented a multiple objective framework to detect communities in complex network for the first time, in which community fitness and community score are minimized and maximized respectively. Gong et al. [15] proposed a multiple objective community detection algorithm based on PSO, of which two evaluation objectives, e.g., Kernel K-Means (KKM) and Ratio Cut (RC), are to be minimized. It introduces decomposition operator to decompose the community detection problem into several scalar problems and then applies the proposed discrete framework to optimize them simultaneously. Shi et al. [12] proposed an evolutionary algorithm called MOCD to detect community structures under a multiple objective framework which optimizes a combination of two negatively correlated objectives. Furthermore, Rahimi et al. [17] improves PSO by modifying particles' movement strategy based on genetic operator, and employs KKM and RC as its objective criteria. The good performance of it was presented in efficiency and quality, nevertheless the normalized mutual information (NMI) criterion used in iteration requires ground-truth communities being given first, which indicated that this method relies on more prior knowledge.

Same as the GA and PSO, pigeon inspired optimization (PIO) algorithm is also an efficient evolutionary computation algorithm. Duan et al. [20] first presented the PIO algorithm and applied to solve air robot path planning problems, in which map and compass operator model is presented based on magnetic field and sun, simultaneously landmark operator model is designed based on landmarks. Inspired by the Pareto sorting scheme, Qiu et al. [21] proposed a variant of PIO named MPIO to solve multi-objective optimization problems. The MPIO merges the map and compass operator with the landmark operator for the navigation of homing pigeons and employs a transition factor to smooth the work transition between the two operators. Improving MPIO based on the hierarchical learning behavior in pigeon flocks, Qiu et al. [22] once again proposed the modified MPIO to coordinate unmanned aerial vehicles fly in a stable formation under complex environments.

In this study, multi-objective pigeon inspired optimization algorithm (MOPIO) is applied to solve the community detection and presents superior performance comparing to the others. MOPIO adjusted the representation and the update of pigeons to adapt optimization problems of community detection through introducing the genetic operator. In this work, Negative Ratio Association (NRA) and Ratio Cut (RC) are employed as objective functions to be minimized. Pareto sorting scheme is utilized to judge non-dominated solutions which are used on later crossover process. A crossover strategy based on global and personal bests is designed, in which a compensation coefficient is developed to stably complete the work transition between the map and compass operator,

and the landmark operator. Besides, it uses the leader selection strategy to determine final result from the optimal solution set. Experiments on real networks validated the good performances of the proposed algorithm.

The remainder of this paper is organized as follows. Section 2 describes the definition of community detection, the related concepts of multi-objective optimization and the definition of original PIO algorithm. Section 3 presents the implementation details of MOPIO. In Section 4, the experimental results are discussed. Finally, conclusions are given in Section 5.

2. Related Works

2.1. Community Detection Problem

A complex network can be modeled as an undirected graph $G = (V, E)$, where V and E denote a set of nodes and a set of edges respectively. A node of the graph can be seen as an entity, while edges denote the relationships among entities. Generally, the topological structure of nodes in complex networks presents a trend of aggregation, which can be referred to as dense clusters or communities. The goal of community detection is to group a set of nodes into dense parts, ensuring that internal connections of a part are denser than connections with other parts [23–25]. Communities in which a node can be a member of more than one community are called overlapping community. On the contrary, the situation of a node can only belong to one community is non-overlapping community. This study focuses on non-overlapping community detection.

Graph G is stored in the form of adjacency matrix defined as A . If there is an edge between node i and node j in the adjacency matrix A , the value of A_{ij} is set to 1, otherwise 0. Since the network is treated as undirected graph, A_{ij} equals to A_{ji} . Given that a community S belongs to the graph G . Let $k_i^{in} = \sum_{j \in S} A_{ij}$ and $k_i^{out} = \sum_{i \in S, j \notin S} A_{ij}$ be the internal and external degree of node i , thus S is a strong community if $\forall i \in S, k_i^{in} > k_i^{out}$, and S is weak if $\sum_{i \in S} k_i^{in} > \sum_{i \in S} k_i^{out}$. That is to say, in a strong community structure, the number of the edges within the community is significantly larger than that of edges between the communities. To sum up, community detection is the process of exploring clusters which gathers nodes.

2.2. Multi-Objective Optimization

For many real applications, such as economy, management, and engineering design, it is difficult to judge the quality of a solution with one measure. Therefore, multi-objective optimization is widely used to solve such problems. Typically, in the process of multi-objective optimization, several complementary objectives are required to measure the quality of a solution and are optimized simultaneously to guide solutions approach to the optimal. The community detection problem can be modeled as an optimization problem [26], and then solved using the multi-objective optimization framework in which a set of solutions that define the best tradeoff among complementary objectives can be obtained. Generally, the multi-objective optimization problem is composed of several objective functions and constraints [27], which can be described as follows:

$$\min_{x \in R^n} F(x) = \min_{x \in R^n} (f_1(x), f_2(x), \dots, f_m(x)) \quad (1)$$

$$s.t. g_i(x) \leq 0, i = 1, 2, \dots, p \quad (2)$$

where $F(x)$ consists of several objective functions that need to be minimized at the same time, $f_m(x)$ is the m_{th} objective function, $X = \{x | x \in R^n, g_i(x) \leq 0, i = 1, 2, \dots, p\}$ is the feasible region of the optimization problem, R^n is the n -dimensional solution space, and $g_i(x)$ is the constraint function.

Pareto scheme is widely applied in multi-objective optimization problem, in which each solution is first assessed according to multiple criteria and a subset of solution to the conditions of Pareto optimality are offered. Below, several terminologies related with Pareto are introduced.

Given two decision vectors $x_1, x_2 \in X$ and x_1 dominating x_2 , they can be written as $x_1 \succ x_2$, if and only if:

$$\forall i \in \{1, 2, \dots, n\}, f_i(x_1) \leq f_i(x_2) \wedge \exists j \in \{1, 2, \dots, n\}, f_j(x_1) < f_j(x_2) \quad (3)$$

If there is no decision vector in the feasible region dominates a decision vector, the vector is called Pareto optimal solution or non-dominated solution. Pareto optimal solution or non-dominated solution is defined as:

$$PS^* = \{x^* \in X | \neg \exists x \in X, x \succ x^*\} \quad (4)$$

Pareto optimality is a situation that no criterion can be better without making at least one criterion worse in a multi-objective optimization problem. For an optimization problem with m objective functions, all Pareto optimal solutions are mapped into a m -dimensional space as points depending on the value of objective functions. The region consisting of these points which respectively corresponding to one solution is named the Pareto optimal front (POF), which is defined as:

$$PF^* = \{F(x^*) = [f_1(x^*), f_2(x^*), \dots, f_n(x^*)]^T | x^* \in PS^*\} \quad (5)$$

2.3. Basic PIO

Solar position, the Earth's magnetic field and landmarks are used by homing pigeons to orient and find nest accurately. Most researchers hold that homing ability is founded on the model of map and compass which rely on the sun and magnetic field, with the map and compass feature enabling pigeons to determine their locations relative to nest for orienting. Besides, pigeons will switch to landmark wayfinding mode about halfway through the journey, and reassess their route for correction. In order to solve the problem of engineering design optimization, Duan et al. [20] proposed a new biologically inspired swarm intelligence algorithm called Pigeon-Inspired Optimization (PIO) for the first time based on the homing behavior of pigeons. By simulating the group behavior of homing pigeons, the map and compass operator model and the landmark operator model are put forward derived from sun and magnetic field, and landmarks, respectively.

For single-objective optimization problems, PIO has achieved superior performance on solving the optimization design problems such as orbital spacecraft formation reconstruction and target detection tasks. So as to fill the gap of PIO in multi-objective optimization research, Multi-objective Pigeon-Inspired Optimization (MPIO) [21] is proposed. PIO uses two independent cycles to simulate the homing characteristics of pigeons, while MPIO merges the map compass operator model and the landmark operator model into an entirety. The work transition between two operators is stably completed with a compensation coefficient introduced, and the Pareto sorting scheme is used to solve multi-objective problems. For a D -dimension search space, in the MPIO, the total number of pigeons with N is randomly initialized. Their positions and velocities are expressed by $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ and $V_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$, respectively, where $i = 1, 2, \dots, N$. The improved location and speed update methods for the next generation of pigeons are as follows:

$$V_i^t = V_i^{t-1} \cdot e^{-R \times t} + rand_1 \cdot tr \cdot (1 - lg_{t_{max}}^t) \cdot (X_{g_{best}} - X_i^{t-1}) + rand_2 \cdot tr \cdot lg_{t_{max}}^t \cdot (X_{center}^{t-1} - X_i^{t-1}) \quad (6)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (7)$$

where t_{max} is the maximum number of iterations and tr is the transition factor. With the increase of t_{max} , individual X is more dependent on X_{center} , than $X_{g_{best}}$. $X_{g_{best}}$ is the best position compared with all pigeon positions during the $t - 1$ iteration of the map compass operator, and X_{center} is a virtual position at the center of pigeon flock corresponding to the landmark operator, that is, the destination to which the pigeon flock will fly. Considering two operators need to be merged and redefined in MPIO, an archive A is set to store the

non-dominated solutions and resolve X_{gbest} and X_{center} . The implementation is introduced in the following discussion.

Through the pareto sorting scheme, the fitness of each pigeon of the current population is evaluated by the established objective function to obtain the non-dominated solution, and then the non-dominated solutions S_1^X in the current generation X are stored in archive A . X_{center} is defined as follows:

$$X_{center}^{t-1} = \frac{\sum_{j=1}^{n_1^X} S_{1j}^X}{n_1^X} \tag{8}$$

The archive A retains the superior non-dominated solutions in S_1^X and removes other bad solutions in the set. X_{gbest} is randomly selected from A .

From the definition of X_{center} in MPIO, this method is not suitable for solving problems on complex network data. As far as community detection problems are concerned, the optimal community partition scheme has nothing to do with the location mean of the non-dominated solution set. In this study, some improvements and innovations have been made based on the MPIO framework. According to the topological characteristics of complex networks, genetic operation is introduced, and the map compass operator model and the landmark operator model are redefined. Corresponding to the two stages, we use personal optimal solutions X_{pbest} and the global optimal solution X_{gbest} to participate in updating of pigeons. The detail of implementation is described in next section.

3. Method

In this section, the multi-objective pigeon inspired optimization for community detection (MOPIO) is described in detail. First, the representation scheme of individual and initialization rules for population used in the MOPIO framework are given, next, two objective functions including Negative Ratio Association (NRA) and Ratio Cut (RC) are described, then Pareto sorting scheme and the search strategy of MOPIO are elaborated; at last, the selection operation for getting an optimal solution from the archive is explained. The flowchart of MOPIO is given in Figure 1.

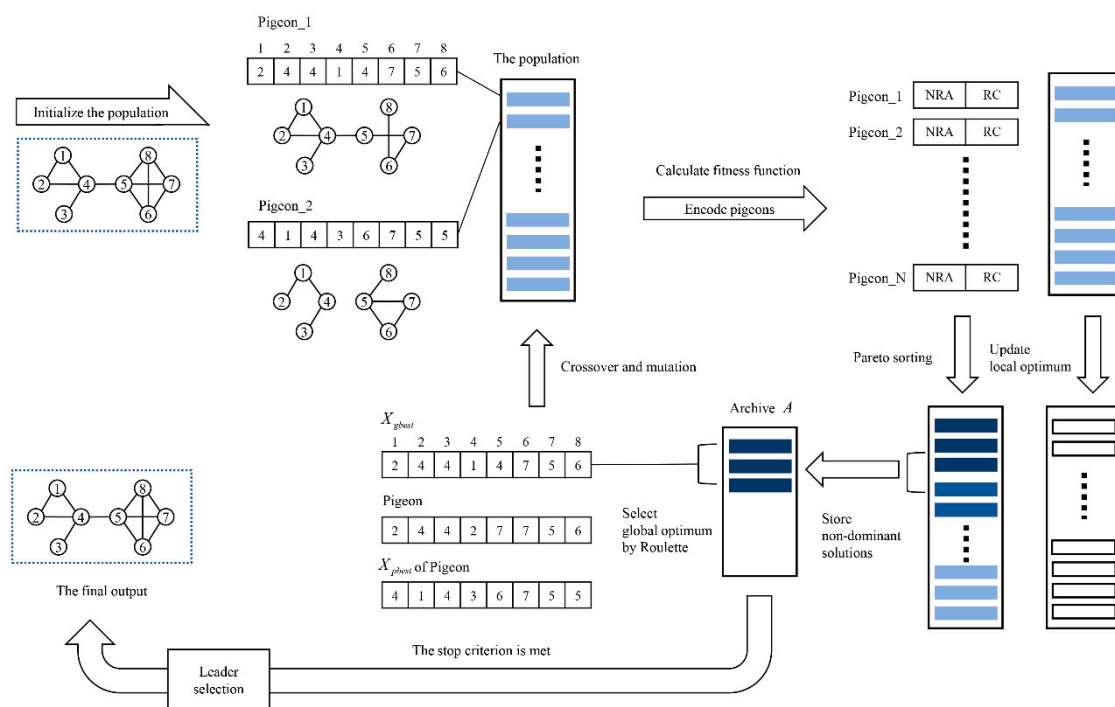


Figure 1. The flowchart of the proposed method MOPIO algorithm.

3.1. Pigeon Representation and Initialization

Considering the adaptability of pigeon inspired optimization for community detection, a pigeon inspired optimization which combines with genetic operators differ from the MPIO is proposed. We described a pigeon in optimization problem through the conception of gene, which is defined by the locus-based adjacency representation (LAR), as well as introduced the crossover and mutation operator instead of original updating operation by velocity. In our method, a pigeon in the population consists of N genes and each gene locus corresponding to a node in the graph possesses a value which is the index of node. For an instance, the value of j for k_{th} gene means there is an edge between node j and node k in this representation. By the decoding operation, a solution can be resolved into a community partition result, in which every connected component is a community. The number of community partition need not to be specified in advance. Moreover, the time consumption of decoding operation is linear, which means that using this representation is efficient.

The initialization operation of population is to randomly select a value from the neighbor nodes of the corresponding node for each locus of pigeon gene, and repeat this operation to initialize the whole pigeon swarm. The LAR scheme can ensure that the number of communities is automatically determined and every individual is a feasible solution, which also provides convenience for the subsequent crossover and mutation operation.

3.2. Fitness Function

In this study, NRA [28] and RC [29] are used to minimize as optimization functions. The NRA is a negative value of RA which measures the density of edges belonging to a same community. A significant community partition corresponds to a high RA value, in which internal edges of each community are dense. In order to facilitate the optimization process, the negative value of RA as one of objective functions. Therefore, NRA indicates the negative value of the sum of the internal edge densities of identified communities, which is calculated as follows:

$$NRA = -RA = -\sum_{i=1}^m \frac{L(C_i, C_i)}{|C_i|} \quad (9)$$

where m represents the number of communities, and the $|C_i|$ is the number of vertices in community i .

Also, RC can be explained as the sum of the density of the links of inter-communities and it is computed as follows:

$$RC = \sum_{i=1}^m \frac{L(C_i, \bar{C}_i)}{|C_i|} \quad (10)$$

where \bar{C}_i is the complementary set of C_i , $\bar{C}_i = D - C_i$, if a group of community structures $D = \{C_1, C_2, \dots, C_m\}$ of G is given. $L(C_i, C_i) = \sum_{j \in C_i, k \in C_i} A_{jk}$ and $L(C_i, \bar{C}_i) = \sum_{j \in C_i, k \in \bar{C}_i} A_{jk}$.

A community partition in which tight connections within communities and sparse connections between communities can be obtained, by minimizing NRA and RC. From the definition of the two objective functions, we can see that minimizing NRA can divide the network into many closely connected communities, but it is easy to create many small communities. Conversely, minimized RC can divide the network into a small number of large communities, which are connected sparsely. Thus, we balance the trade-off between them by multi-objective optimization method based on Pareto scheme to achieve the purpose of community detection.

3.3. Pareto Sorting Scheme

The Pareto sorting scheme [30] is used in the MOPIO algorithm with an elite individual candidate archive to maintain the non-dominated solutions. Pareto sorting occurs after

the update operation of individuals. According to the comparison of the value of objective functions among individuals, the dominant relationship among individuals is determined, and the solutions in a dominant side will be reserved. The dominance relationship has been described in Section 2.2. For updating the archive A , solutions reserved above are compared with those original solutions in the archive to maintain non-dominated ones. Finally, the crowding distance between adjacent solutions is calculated, solutions ranking in descending order of fitness. On the basis of the sum of crowding distances in different criteria, all solutions are ranked in descending order again. The crowding distance is defined as follows:

$$Dis(x_i) = \sum_{k=1}^m \frac{f_k(x_{j+1}^{(k)}) - f_k(x_{j-1}^{(k)})}{f_k^{\max} - f_k^{\min}} \quad (11)$$

where x_i is the i_{th} solution in the archive, $x_{j+1}^{(k)}$ represents the previous solution of x_i when the solutions in the archive are sorted according to the descending order of k_{th} objective function, that is to say, x_i ranks j_{th} when sorting according to the descending order of k_{th} objective function. The maximum and minimum values of the k_{th} objective functions are f_k^{\max} and f_k^{\min} , respectively. To ensure the diversity of solutions, it is considered that the larger crowding distance means better. And the global optimal solution is selected from the archive, which is described in the next section.

3.4. Search Strategy

Search phase of MOPIO is achieved by pigeons learning from non-dominated individuals. Learning process of a pigeon is composed of itself, personal optimums, and the global optimum of population. At the initial search phase, each pigeon will learn more about its own experience. As the number of iterations increases, pigeons will learn more from the global optimum of population. The improved update strategy based on two models within PIO, which makes the proposed method more suitable for solving optimization problems in community detection. In this study, the map and compass operator is merged with the landmark operator in a different way from MPIO, meanwhile, genetic operator including crossover and mutation is introduced. The detailed operator strategy is explained as follows.

3.4.1. Optimal Solution Selecting Strategy

MPIO is a method proposed for the design of mechanical parameter, in which movement of pigeons are adjusted by velocity update strategy depending on two operators, X_{center} and X_{gbest} . In view of the characteristics of the discretization of network data in the problem of community detection, this paper proposes a novel update strategy based on crossover and mutation to replace velocity-based strategy. Correspondingly, the strategy for selecting individuals with high fitness is designed to determine the targets that pigeons inherit from. The genetic operator proposed by MOPIO is completed by a pigeon and the personal optimal solution and the global optimal solution. The personal optimal solution is the optimal solution exploited by a pigeon in its own iteration process, and the global optimal solution is a certain solution selected from the archive, both can inherit a part of the gene fragments in crossover phase by the pigeon corresponding to the personal optimal solution. The roulette wheel is used as a global optimal solution selection strategy, which is executed after the non-dominated solution set is arranged in descending order according to crowding distance. However, the optimal individual selection strategy we adopted is different from the general situation if and only if the population is first generation. In the first iteration, the initial state of the archive A is empty. The initial state of each pigeon of population is recorded as the personal optimal solution which will participate in the update operation in the next generation. For the global optimal solution, with non-dominated sorting scheme performed in the initial population, the pigeons are sorted according to the non-dominated rank, and each solution is compared with all the other solutions to check whether it is dominated. A set of non-dominated solutions identified by above operation are stored in the archive A . After calculating the

crowding distance between solution in A and assigning weights to the pigeons with the calculated values, a pigeon is selected from A using roulette method as the global optimal solution.

3.4.2. Crossover and Mutation

The update operation is carried out after both personal optimal and the global optimal had been determined. Each pigeon can inherit better gene fragments from the two optimal solutions with higher fitness to produce offspring, which is achieved through a multi-individual crossover operation. The detail of crossover and mutation operation is depicted in Figure 2.

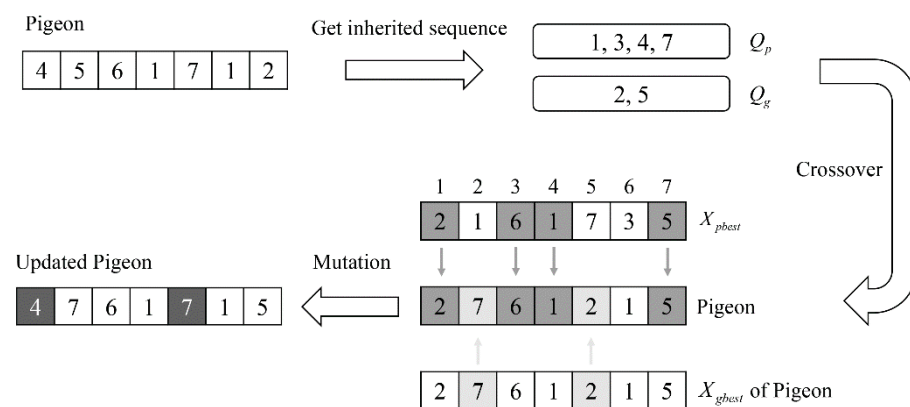


Figure 2. Crossover and mutation operation.

To perform the crossover operation, firstly, two random sequences corresponding to the personal optimal solution and the global optimal solution are generated whose values range from $[0, N]$. N is the dimension of the problem, which is also the length of the gene that represents the pigeon state. For each random sequence, they indicate indices of genes that will be inherited, and the uniqueness of indices is guaranteed. As shown in Figure 2, Q_p refers to the index of genes to be inherited from the personal optimal solution, and Q_g refers to the index of genes to be inherited from the global optimal solution. The numbers of gene segments inherited respectively from two solutions are related to the number of current iterations. The definition of gene length to be inherited is as follows:

$$length_p = (1 - \log_{tmax}^t) \cdot pc \cdot N \quad (12)$$

$$length_g = \log_{tmax}^t \cdot pc \cdot N \quad (13)$$

where pc is crossover probability, $tmax$ is the maximum number of iterations, and t is the number of the current iteration. The mutation strategy is to make a pigeon randomly select a neighbor node as a new gene with probability pm for each gene locus. It can be seen from the definition that, at the early generation, more gene fragments can be obtained from the personal optimal solution through update operation. As the number of iterations increases, the preference for inherited gene fragments gradually tends to the global optimal solution. In this way, the population richness at the beginning of search phase can be guaranteed, and the convergence speed of the algorithm at the end of search phase can be accelerated.

3.5. Leader Selection Operation

When the termination criteria were met, the optimal solution selection operation would be performed on the archive to determine the final output of the algorithm. For selecting the result from the non-dominated solutions in the archive A , the leader selection operation is designed. First, the set of solutions in A is sorted in descending according to crowding distance, the reciprocal of each solution's ranking is recorded as its crowding distance score. Then, the modularity of each solution is calculated, similarly, the reciprocal

of the ranking of each solution in descending order is recorded as its modularity score. After calculating the total score of crowding distance and the modularity of each solution, the final result is determined by the roulette method, in which the solution with a higher score has a higher roulette weight. Meanwhile, a preferred ratio p is set to remove some individuals for eliminating the influence of the solution with too large value of single objective function. The solution with as large crowding distance and modularity as possible is selected for balancing the trade-off between crowding distance and modularity.

4. Results and Discussion

In this section, the experiments of MOPIO were conducted on four popular real-world networks, i.e., the Zachary's karate club [31], FB50 [32], the American College Football [5] and the Krebs' books on US politics [33]. To evaluate the performance of MOPIO, the comparison with four state-of-the-art models, such as MOGA-Net [11], MOPSO [17], FN [34] and Meme-Net [8], were implemented. Considering that the parameter setting is a challenging problem for evolutionary algorithms, the method of trial and error was adopted, which is reasonable to choose the value that performs well in our experiment. Based on this method, the parameters of MOPIO are presented in Table 1. Meanwhile, population size and iterations of all evolutionary algorithms in the comparison method are consistent with the proposed method in this study, and other parameters use the recommended parameters in their own method. It is worth noting that all the reported results in the experiments are average values obtained from 20 runs of each algorithm.

Table 1. Algorithm parameters.

N	Population Size	100
I	The number of MOPIO iteration	50
pc	crossover probability	0.8
pm	mutation probability	0.4
p	preferred ratio	0.25

4.1. Evaluation Metrics

Two commonly used evaluation metrics, i.e., the Normalized Mutual Information (NMI) [35] and the Adjusted Rand Index (ARI) [36], were adopted to estimate the quality of the partitions in the experiments. NMI is a metric used to measure the distribution similarity between community partitions identified by community detection algorithms and real community partitions. ARI is another widely recognized metric for evaluating the similarity between two partitions. We consider that NMI and ARI are common measures to evaluate the performance of community detection algorithms, and whether the ground truth clustering is balanced will lead to different NMI and ARI values. Therefore, we use these two measures to evaluate the experimental results comprehensively.

Given that two community partition, P_1 and P_2 , correspond to real partition and detected partition respectively, the NMI is defined as:

$$NMI(P_1, P_2) = \frac{-2 \sum_{i=1}^{C_{P_1}} \sum_{j=1}^{C_{P_2}} C_{ij} \log(C_{ij}N / C_i \cdot C_{.j})}{\sum_{i=1}^{C_{P_1}} C_i \cdot \log(C_i / N) + \sum_{j=1}^{C_{P_2}} C_j \cdot \log(C_j / N)} \quad (14)$$

where C is the confusion matrix of classification results between real partition and experimental partition. C_{ij} represents the number of nodes belonging to both community i in the P_1 and community j in the P_2 , and $C_i \cdot (C_{.j})$ is the sum of elements of C in row i (column j). $C_{P_1}(C_{P_2})$ and N represent the number of communities of $P_1(P_2)$ and the total number of nodes, respectively. The NMI value range from 0 to 1, The larger the NMI value, the higher the similarity between P_1 and P_2 .

ARI, another common evaluation function of clustering result, is revised by RI, and RI is defined as follows:

$$RI = \frac{TP + TN}{TP + FN + FP + TN} \quad (15)$$

where TP represents the number of node pairs belonging to the same community in P_1 and P_2 ; FN represents the number of node pairs that belong to the same community in P_1 but different communities in P_2 ; Contrasting to FN , FP represents the number of node pairs that belong to different communities in P_1 but the same community in P_2 ; TN represents the number of node pairs divided into different communities in P_1 and P_2 .

Considering that the RI value is not close to 0, causing the lower degree discrimination in clustering results, ARI was introduced to modify the shortcomings. And ARI is defined as follows:

$$ARI = \frac{RI - E(RI)}{MAX(RI) - E(RI)} \quad (16)$$

where $E(RI)$ is the expected value of RI and $MAX(RI)$ is the maximum value of RI .

Furthermore, three classic measures, including precision, recall and F-measure, were adopted to evaluate performance of MOPIO. Precision is the ratio of true-positive predictions out of all positive predictions, and Recall is the ratio of true-positive predictions to all true predictions, which can be defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (17)$$

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

F-measure is the harmonic average of accuracy and recall, which defines as follows:

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (19)$$

4.2. Experimental Results

In this section, experimental results are presented on real-world network datasets, i.e., the Zachary's karate club, FB50, the American College Football and the Krebs' books on US politics. The characteristics of the networks are shown in Table 2.

Table 2. The characteristics of the networks.

Network	Node	Edge	Community
Zachary's karate club	34	78	2
FB50	50	404	4
American College Football	115	613	12
Krebs' books on US politics	105	441	3

Table 3 shows the maximum and average of NMI and ARI for all comparison methods in 20 independent runs. As can be seen from the first part of Table 1, which is results on first dataset, maximum value of NMI obtained from MOPIO and Meme-Net reaches 1.000. The same is true of maximum ARI, indicating that both methods can search for standard community division results. Average NMI of MOPIO is higher than that of all other methods, meaning that the experimental results of MOPIO on karate network are more stable. In addition, the maximum NMI detected by MOPSO is lower than that of our method and Meme-Net, but it is quite close to 1.000. And the average NMI of the former is slightly higher than that of Meme-Net, so the performance of MOPSO and Meme-Net may be similar on the karate network. However, by comparing the maximum and average ARI values of the two methods, we can see that the classification results of Meme-Net are better. The true structure of the Zachary's karate club network with two real partitions (blue and

brown) is given in Figure 3a. Other panels in Figure 3 show results with maximum NMI detected by the five methods. Nodes of the same color belong to the same community. The results of MOPIO and Meme-net are consistent with the benchmark network, that of MOPSO is similar to the benchmark network, and the remaining methods identified more than two communities.

Table 3. Results obtained by the four algorithms on real-world network.

Dataset	Metrics	MOPIO	MOPSO	MOGA-Net	FN	Meme-Net
Karate	NMI (max)	1.000	0.930	0.707	0.692	1.000
	NMI (avg)	0.860	0.556	0.628	0.692	0.501
	ARI (max)	1.000	0.882	0.416	0.680	1.000
	ARI (avg)	0.856	0.467	0.415	0.680	0.477
Football	NMI (max)	0.816	0.399	0.800	0.726	0.887
	NMI (avg)	0.754	0.122	0.762	0.726	0.795
	ARI (max)	0.670	0.113	0.629	0.491	0.744
	ARI (avg)	0.573	0.045	0.485	0.491	0.581
Fb50	NMI (max)	1.000	0.902	0.938	0.938	1.000
	NMI (avg)	1.000	0.794	0.938	0.938	0.997
	ARI (max)	1.000	0.814	0.954	0.954	1.000
	ARI (avg)	1.000	0.580	0.954	0.954	0.998
Polbooks	NMI (max)	0.606	0.456	0.564	0.516	0.574
	NMI (avg)	0.494	0.163	0.524	0.516	0.427
	ARI (max)	0.709	0.248	0.665	0.609	0.675
	ARI (avg)	0.559	0.080	0.579	0.609	0.434

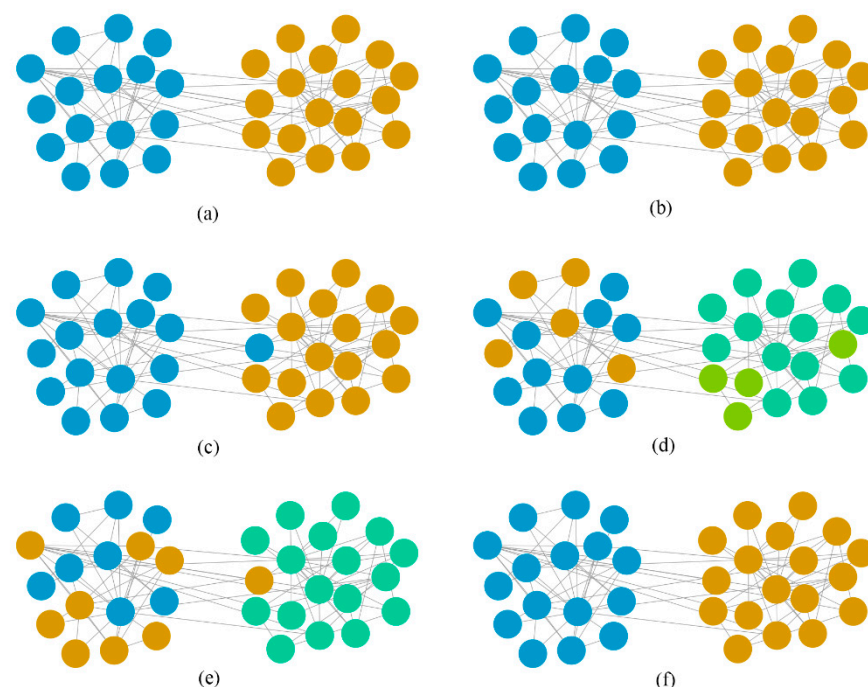


Figure 3. True structure of the Zachary's karate club network and results detected by five methods. (a) Benchmark network; (b) MOPIO; (c) MOPSO; (d) MOGA-Net; (e) FN; (f) Meme-Net.

The second part of Table 1 is the results of American College Football network. Whether it is the maximum and average values of NMI or those of ARI, Meme-Net is better than any other methods. Furthermore, the experimental performance of our method, ranks second, is close to Meme-Net, and it also performs well on the American College Football network. Networks depicted in Figure 4 are results with maximum NMI value

detected by above methods and true partition of American College Football network. From the grouping of node colors in Figure 4, Meme-Net shows the closest result to the benchmark network, and MOPIO is closely behind. Except for MOPSO, most methods can detect structures like but not the same as the benchmark network.

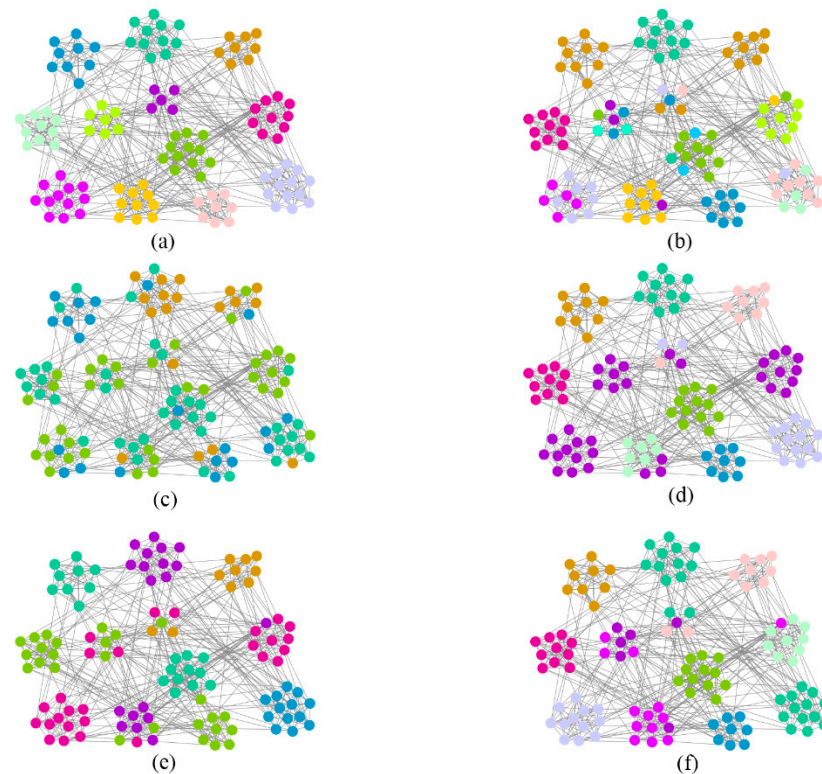


Figure 4. True structure of American College Football network and results detected by five methods. (a) Benchmark network; (b) MOPIO; (c) MOPSO; (d) MOGA-Net; (e) FN; (f) Meme-Net.

In the experimental results on the FB50 data set, the maximum NMI and ARI of MOPIO is 1.000, which is better than all methods. The average values of NMI and ARI indicate that MOPIO can search out the standard partition scheme stably and accurately. Meme-Net ranks second and is slightly lower than MOPIO in terms of stability. MOGA-Net and FN have the same performance on FB50 data, so we can see that they can stably detect the same partition in 20 independent runs. However, this kind of partition slightly differs from the standard partition. Part of the experimental results are shown in Figure 5.

In the comparison of Krebs' books on US politics network results, experimental results are similar to the case on football dataset, in which the result consistent with standard partition have not been found. The politics network is extraordinary complex so that all the comparison methods perform poorly on the dataset, as depicted in Figure 6. The best results of the maximum NMI and ARI are obtained from MOPIO, which are 0.606 and 0.709, respectively.

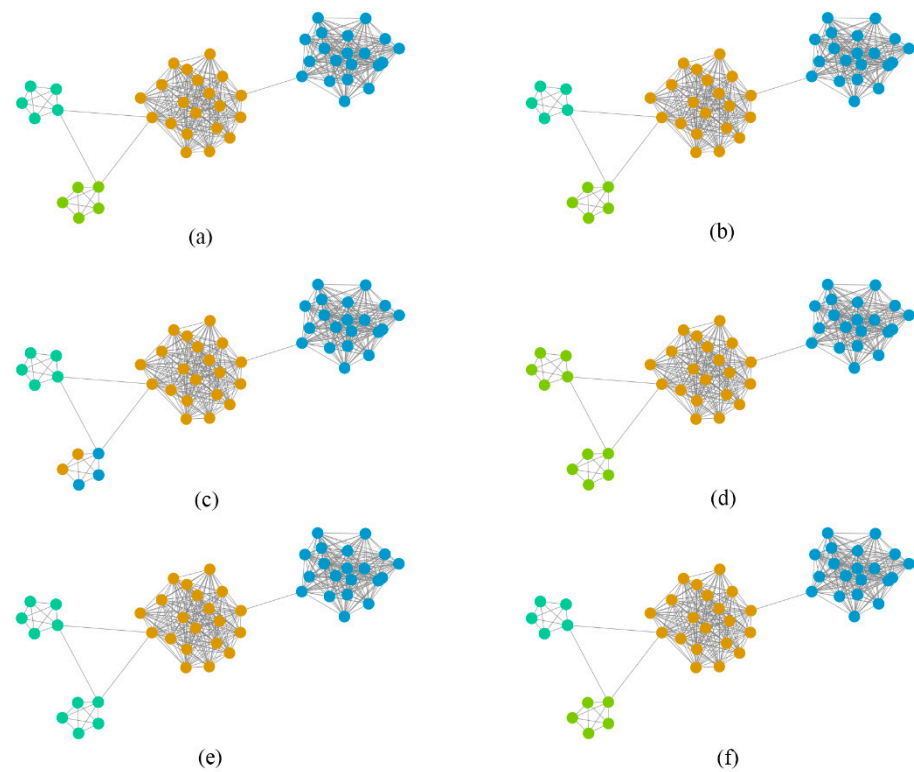


Figure 5. True structure of FB50 and results detected by five methods. (a) Benchmark network; (b) MOPIO; (c) MOPSO; (d) MOGA-Net; (e) FN; (f) Meme-Net.

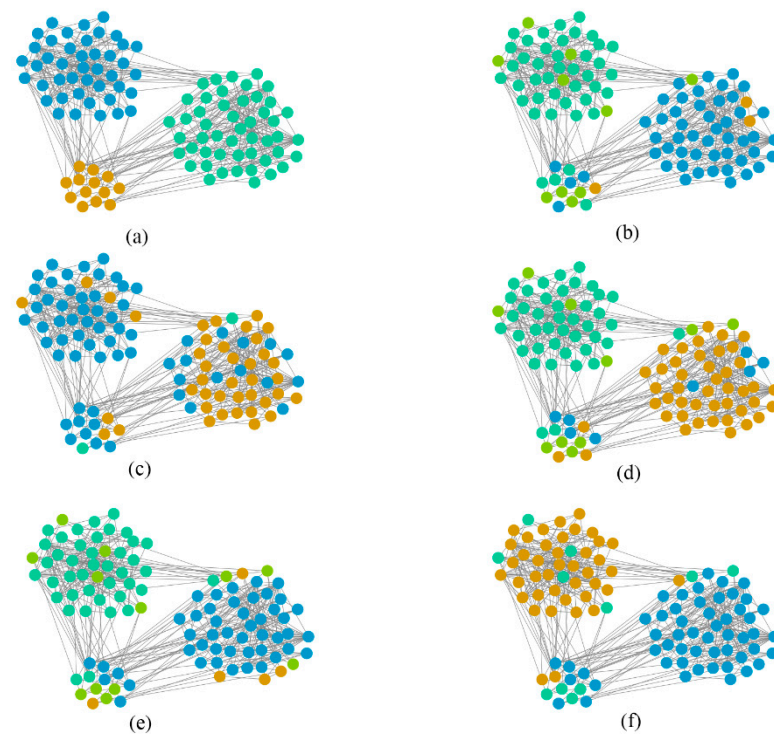


Figure 6. True structure of Krebs' books on US politics network and results detected by five methods. (a) Benchmark network; (b) MOPIO; (c) MOPSO; (d) MOGA-Net; (e) FN; (f) Meme-Net.

To analyze their classification performance, we also calculate the Precision, Recall and F-measure of the proposed method as well as those of four comparison methods. The

experimental results are shown in Table 4. The letters P, R and F correspond to the results of Precision, Recall, F-measure in turn.

Table 4. Average results of Precision, Recall and F-measure from 20 independent runs.

Dataset	Metrics	MOPIO	MOPSO	MOGA-Net	FN	Meme-Net
Karate	P	<u>0.624</u>	0.631	0.229	0.370	0.402
	R	<u>0.576</u>	0.612	0.172	0.185	0.453
	F	<u>0.596</u>	0.615	0.196	0.247	0.410
Football	P	0.164	0.028	0.110	0.097	0.101
	R	0.182	0.108	0.150	0.165	0.142
	F	0.160	0.040	0.118	0.119	0.108
Fb50	P	1.000	0.462	0.625	0.625	0.981
	R	1.000	0.538	0.750	0.750	0.988
	F	1.000	0.492	0.667	0.667	0.983
Polbooks	P	0.095	0.209	0.052	0.056	0.123
	R	0.080	0.359	0.054	0.024	0.104
	F	0.074	0.237	0.033	0.033	0.085

The classification performance of our approach MOPIO is superior to all other methods on FB50 and the American College Football, MOPSO also outperforms other methods on two data, the Zachary's karate club and the Krebs' books on US politics. However, it is easy to see that the overall experimental performance of MOPSO in Table 3 is poor, except for good results in the karate network. The optimization of NRA and RC in MOPIO's iteration tends to result in more communities. The Krebs' books on US politics network is essentially a network with low modularity, and the experiment of MOPIO on this dataset will obtain the results containing more than three communities in most cases. This has powerful influence on precision, recall and F-measure calculated by macro average rule. We hope that the trade-off between NRA and RC can be better balanced in the future work.

To sum up, the experimental results show that MOPIO can perform well in terms of search accuracy and stability on the real data with standard community partition. With the increase of number of iterations, the proportion of learning from global and personal optimal individuals changes dynamically, so that the algorithm can explore the solution space sufficiently at the initial search phase and guarantee high convergent precision at the end of search phase, which ensures accuracy and stability of the results.

5. Conclusions

In this paper, a community detection method named MOPIO has been proposed, whose contribution mainly lies in an update strategy based on multi-individual crossover and an improved PIO scheme for community detection. After adopting the compensation coefficient in this strategy, the source of gene fragment will tend to the global optimal solution rather than the personal optimal solution as the number of iterations increases. In addition, this optimized update strategy, the pigeon inspired optimization method and the Pareto sorting scheme are combined into a community structure detection framework in MOPIO. Experiments show that the performance of MOPIO is generally better than the other four methods on the real network data set, which shows that MOPIO is promising for detecting real community structure. MOPIO is implemented in python and is freely available from <https://github.com/CDMB-lab/MOPIO>.

The advantage of MOPIO is to ensure population diversity and the adequacy of exploring for solution space at the beginning of the search phase, and to guarantee high convergence precision to obtain community partitions close to the real structure. However, MOPIO still has the following limitations, which is worthy of our further study and exploration. The method focuses on searching the partition consistent with the standard division of real network data, which often does not correspond to the best modularity. Therefore, our method has some difficulties for the artificial networks that merely rely on

modular optimization for detection without standard division. In the future, optimizing the experimental framework and analysis method of community detection is our goal, which is a research direction worthy of attention.

Author Contributions: Conceptualization, J.S.; methodology, Y.L. and J.-X.L.; validation, J.S., Y.S. and Y.Z.; software, Y.L.; formal analysis, J.S.; writing—original draft preparation, Y.L.; writing—review and editing, J.S., Y.L. and F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China (61972226, 31872242, 61872220, and 61902216) and the China Postdoctoral Science Foundation (2018M642635).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We are grateful to the anonymous reviewers whose suggestions and comments contributed to the significant improvement of this paper.

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

References

- Ding, Q.; Shang, J.; Sun, Y.; Wang, X.; Liu, J.X. HC-HDSD: A method of hypergraph construction and high-density subgraph detection for inferring high-order epistatic interactions. *Comput. Biol. Chem.* **2019**, *78*, 440–447. [\[CrossRef\]](#)
- Radanliev, P.; De Roure, D.C.; Nurse, J.R.C.; Mantilla Montalvo, R.; Cannady, S.; Santos, O.; Maddox, L.T.; Burnap, P.; Maple, C. Future developments in standardisation of cyber risk in the Internet of Things (IoT). *Sn Appl. Sci.* **2020**, *2*, 169. [\[CrossRef\]](#)
- Cai, Q.; Ma, L.; Gong, M. A survey on network community detection based on evolutionary computation. *Int. J. Bio Inspired Comput.* **2014**, *8*, 84–98. [\[CrossRef\]](#)
- Ye, X.; Fei, C. Researches on Evaluations of Large-scale Complex Networks Topologies. *Procedia Comput. Ence* **2017**, *107*, 577–583. [\[CrossRef\]](#)
- Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [\[CrossRef\]](#)
- Li, X.; Wu, X.; Xu, S.; Qing, S.; Chang, P. A novel complex network community detection approach using discrete particle swarm optimization with particle diversity and mutation. *Appl. Soft Comput.* **2019**, *81*, 105476. [\[CrossRef\]](#)
- Pizzuti, C. Evolutionary Computation for Community Detection in Networks: A Review. *IEEE Trans. Evol. Comput.* **2018**, *22*, 464–483. [\[CrossRef\]](#)
- Gong, M.; Fu, B.; Jiao, L.; Du, H. Memetic algorithm for community detection in networks. *Phys. Rev. E* **2011**, *84*, 056101. [\[CrossRef\]](#)
- Zhang, C.; Hei, X.; Yang, D.; Wang, L. A Memetic Particle Swarm Optimization Algorithm for Community Detection in Complex Networks. *Int. J. Pattern Recognit. Artif. Intell.* **2016**, *30*, 1659003. [\[CrossRef\]](#)
- Guo, X.; Su, J.; Zhou, H.; Liu, C.; Cao, J.; Li, L. Community Detection Based on Genetic Algorithm Using Local Structural Similarity. *IEEE Access* **2019**, *7*, 134583–134600. [\[CrossRef\]](#)
- Pizzuti, C. A Multiobjective Genetic Algorithm to Find Communities in Complex Networks. *IEEE Trans. Evol. Comput.* **2012**, *16*, 418–430. [\[CrossRef\]](#)
- Shi, C.; Yan, Z.; Cai, Y.; Wu, B. Multi-objective community detection in complex networks. *Appl. Soft Comput.* **2012**, *12*, 850–859. [\[CrossRef\]](#)
- Amiri, B.; Hossain, L.; Crawford, J.W.; Wigand, R.T. Community Detection in Complex Networks: Multi-objective Enhanced Firefly Algorithm. *Knowl. Based Syst.* **2013**, *46*, 1–11. [\[CrossRef\]](#)
- Cai, Q.; Gong, M.; Shen, B.; Ma, L.; Jiao, L. Discrete particle swarm optimization for identifying community structures in signed social networks. *Neural Netw.* **2014**, *58*, 4–13. [\[CrossRef\]](#) [\[PubMed\]](#)
- Gong, M.; Cai, Q.; Chen, X.; Ma, L. Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition. *IEEE Trans. Evol. Comput.* **2014**, *18*, 82–97. [\[CrossRef\]](#)
- Zhou, D.; Wang, X. A Neighborhood-Impact Based Community Detection Algorithm via Discrete PSO. *Math. Probl. Eng.* **2016**, *2016*, 3790590. [\[CrossRef\]](#)
- Rahimi, S.; Abdollahpouri, A.; Moradi, P. A multi-objective particle swarm optimization algorithm for community detection in complex networks. *Swarm Evol. Comput.* **2017**, *39*, 297–309. [\[CrossRef\]](#)
- Mu, C.; Zhang, J.; Liu, Y.; Qu, R.; Huang, T. Multi-objective ant colony optimization algorithm based on decomposition for community detection in complex networks. *Soft Comput.* **2019**, *23*, 12683–12709. [\[CrossRef\]](#)
- Liu, X.; Du, Y.; Jiang, M.; Zeng, X. Multiobjective Particle Swarm Optimization Based on Network Embedding for Complex Network Community Detection. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 1–13. [\[CrossRef\]](#)

20. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [[CrossRef](#)]
21. Qiu, H.; Duan, H. Multi-objective pigeon-inspired optimization for brushless direct current motor parameter design. *Sci. China Technol. Sci.* **2015**, *58*, 1915–1923. [[CrossRef](#)]
22. Qiu, H.; Duan, H. A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles. *Inf. Ences* **2020**, *509*, 515–529. [[CrossRef](#)]
23. Newman, M.E.J.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. [[CrossRef](#)]
24. Radicchi, F.; Castellano, C.; Cecconi, F.; Loreto, V.; Parisi, D. Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 2658–2663. [[CrossRef](#)]
25. Pourkazemi, M.; Keyvanpour, M.R. Community detection in social network by using a multi-objective evolutionary algorithm. *Intell. Data Anal.* **2017**, *21*, 385–409. [[CrossRef](#)]
26. Handl, J.; Knowles, J. An Evolutionary Approach to Multiobjective Clustering. *IEEE Trans. Evol. Comput.* **2007**, *11*, 56–76. [[CrossRef](#)]
27. Gong, M.; Jiao, L.; Du, H.; Bo, L. Multiobjective Immune Algorithm with Nondominated Neighbor-Based Selection. *Evol. Comput.* **2008**, *16*, 225–255. [[CrossRef](#)]
28. Angelini, L.; Boccaletti, S.; Marinazzo, D.; Pellicoro, M.; Stramaglia, S. Identification of network modules by optimization of ratio association. *Chaos* **2007**, *17*, 023114. [[CrossRef](#)]
29. Wei, Y.C.; Cheng, C. Ratio cut partitioning for hierarchical designs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **1991**, *10*, 911–921. [[CrossRef](#)]
30. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
31. Zachary, W.W. An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropol. Res.* **1977**, *33*, 452–473. [[CrossRef](#)]
32. Fortunato, S.; Barthélemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 36–41. [[CrossRef](#)] [[PubMed](#)]
33. Newman, M.E.J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [[CrossRef](#)] [[PubMed](#)]
34. Newman, M.E.J. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *69*, 066133. [[CrossRef](#)]
35. Danon, L.; Diazguilera, A.; Duch, J.; Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, *2005*, 09008. [[CrossRef](#)]
36. Zhang, S.; Wong, H.; Shen, Y. Generalized Adjusted Rand Indices for cluster ensembles. *Pattern Recognit.* **2012**, *45*, 2214–2226. [[CrossRef](#)]