

## Article

# A Multi-Objective Pigeon-Inspired Optimization Algorithm for Community Detection in Complex Networks

Lin Yu <sup>1,†</sup> , Xiaodan Guo <sup>2</sup>, Dongdong Zhou <sup>2</sup> and Jie Zhang <sup>1,\*,†</sup> <sup>1</sup> School of Automation, Nanjing University of Science and Technology, Xiaolingwei Street, Nanjing 210094, China; yulin@njjust.edu.cn<sup>2</sup> Northern Information Control Research Institute Group Co., Jiangjun Street, Nanjing 211153, China

\* Correspondence: jiezhangnjjust@163.com

† These authors contributed equally to this work.

**Abstract:** Community structure is a very interesting attribute and feature in complex networks, which has attracted scholars' attention and research on community detection. Many single-objective optimization algorithms have been migrated and modified to serve community detection problems. Due to the limitation of resolution, the final algorithm implementation effect is not ideal. In this paper, a multi-objective community detection method based on a pigeon-inspired optimization algorithm, MOPIO-Net, is proposed. Firstly, the PIO algorithm is discretized in terms of the solution space representation, position, and velocity-updating strategies to adapt to discrete community detection scenarios. Secondly, by minimizing the two objective functions of community score and community fitness at the same time, the community structure with a tight interior and sparse exterior is obtained. Finally, for the misclassification caused by boundary nodes, a mutation strategy is added to improve the accuracy of the final community recognition. Experiments on synthetic and real networks verify that the proposed algorithm is more accurate in community recognition compared to 11 benchmark algorithms, confirming the effectiveness of the proposed method.

**Keywords:** complex networks; community detection; heuristic algorithm; pigeon-inspired optimization; multi-objective optimization

**MSC:** 05C82

**Citation:** Yu, L.; Guo, X.; Zhou, D.; Zhang, J. A Multi-Objective Pigeon-Inspired Optimization Algorithm for Community Detection in Complex Networks. *Mathematics* **2024**, *12*, 1486. <https://doi.org/10.3390/math12101486>

Academic Editors: Janusz Miśkiewicz and Giulia Rotundo

Received: 19 April 2024

Revised: 30 April 2024

Accepted: 7 May 2024

Published: 10 May 2024



**Copyright:** © 2024 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

Networks are used to represent various types of complex systems in many fields such as computer science, physics and mathematics [1]. Several common complex systems include biological networks [2], social networks [3], information networks [4], and so on. Complex networks can reveal some potential rules and features, such as the community structure. In the social field, community detection of social networks can discover friends with common hobbies and interests. In the biological protein networks, community detection can uncover proteins with the same function, which is of great significance for biological gene repair.

From the perspective of graph theory, a network is a graph that can be infinitely large or infinitely small, where the vertices in the graph represent the objects in the network, and the edges represent the direct relationships between the objects. A community is defined as a subset of nodes in the graph that are closely connected to each other, while the nodes between communities are sparsely connected. This characteristic of community structure drives scholars in various fields to conduct research. Community detection has been widely used in social relationship analysis [5], recommendation systems [6,7], link prediction [8], and virus transmission [9,10].

In recent years, many community detection methods have been proposed. In 2002, Girvan et al. proposed the GN algorithm [11] to obtain the community structure in the net-

work by continuously removing the edges with the highest edge mediators. Newman [12] introduced the concept of modularity, which allows community detection to be modeled as an NP-hard optimization problem. Subsequently, more and more metrics have been proposed to assess the quality of community detection, such as community fitness [13] and community score [14]. Benefiting from these metrics, a great number of algorithms based on intelligent optimization are used to solve the community detection problem. Pizzuti [14] proposed a genetic algorithm for community detection to obtain the best division results by optimizing the objective function community score; Li [15] designed an extended compact genetic algorithm using modularity as the optimization objective for community detection; Gong et al. proposed a memetic algorithm based on community detection, called Meme-Net, using module density as an optimization criterion [16]. All the above-mentioned papers are optimized for only one metric criterion, and satisfactory results are achieved over the GN algorithm and the FN algorithm. However, the literature [17] suggested that solving the community detection problem based on single-objective optimization is flawed, which is the resolution problem. Single-objective optimization attempts to find the larger communities in the network, ignoring the small communities that really exist in the network. In addition, the metrics do not fully reveal this characteristic of community structure. There are metrics that attempt to strengthen intra-community connections and metrics that attempt to weaken inter-community connections. Single-objective optimization does not allow for trade-offs between multiple metrics [18].

In response to the above reasons, scholars have started to try to use multi-objective optimization to weigh multiple conflicting metrics to improve the accuracy of community delineation. Pizzuti proposed a multi-objective genetic algorithm for solving community detection, which is known as the MOGA-Net algorithm [19]. Rahimi employed a discrete particle swarm algorithm to optimize the community structure using multi-objective optimization as a framework [20]. Messaoudi proposed a multi-objective bat-based optimization algorithm for the dynamic community detection problem [21]. Li designed an adaptive evolutionary algorithm to extract communities in the network [22]. Chen [23] proposed the MODTLBO/D algorithm for community detection based on a multi-objective teaching–learning-based optimization algorithm combined with a decomposition mechanism. Ji [24] integrated the weighted simulated annealing local search operator into multi-objective ant colony optimization to expand the search range and introduce decomposition mechanisms to enhance the accuracy of community detection. Li [25] designed a decomposition-based multi-objective chemical reaction optimization algorithm to improve the efficiency of community mining with the help of dynamic changes in the population of the algorithm. The authors of [26] proposed a metaheuristic approach based on a variable neighborhood search, which leverages the combination of quality and diversity of a constructive procedure inspired by a greedy randomized adaptation procedure for detecting communities. Meanwhile, there are many scholars who have made outstanding contributions in the field of multi-objective community detection. Ma [27] proposed a two-stage multi-objective community detection algorithm with local search and global search to merge local communities through a boundary control strategy. In the literature [28], a new optimization objective, namely “balanced modularity”, is introduced. Liu [29] introduced network embedding to map nodes to a low-dimensional space, which effectively reduces the search space through a consensus propagation strategy. Pizzuti [30] proposed a multi-objective genetic framework, which integrates the topological and compositional dimensions to uncover community structure in attributed networks. The approach allows for the experimentation of different structural measures to search for densely connected communities and similarity measures between attributes to obtain high intracommunity feature homogeneity. In the literature [31], the Grey Wolf optimization algorithm and the Label Propagation algorithm were improved and combined for better performance.

A community detection algorithm based on the multi-objective pigeon-inspired optimization algorithm was proposed, and the contribution of our work consists of three main aspects:

- (1) We utilize the excellent optimization capabilities of the pigeon-inspired optimization algorithm and combine it with a multi-objective optimization strategy to form a novel algorithm for community detection problems in a complex network.
- (2) We have re-discretized the pigeon-inspired optimization algorithm for the community detection problem. The velocity and position update formulas applicable to the community structure representation are redefined.
- (3) We provide the definition of a boundary node. The misclassification of boundary nodes is a key factor affecting community detection. The corresponding variation strategies are proposed for boundary nodes and non-boundary nodes to improve the accuracy of community partitioning.

## 2. Background and Related Works

### 2.1. Community Definition

The definition of community is unclear [32]. There is a generally accepted consensus that a community is a subset of different nodes, with tightly connected nodes within the set and sparsely connected nodes between the sets [33]. Nodes form communities among themselves based on functional or other shared characteristics.

A network is usually represented in the form of an undirected graph:

$$G = (V, E) \quad (1)$$

where  $V$  represents the vertices in the network and  $E$  is described as the connection between two vertices in the network. From a mathematical point of view, a network can be represented in terms of an adjacency matrix  $A = \{V_i, V_j\}, i, j = 1, 2, \dots, N$ .  $N$  denotes the number of nodes in the network. Where there is a real connection between  $V_i$  and  $V_j$ ,  $A_{ij} = 1$ .  $V_i$  and  $V_j$  are neighbor nodes to each other, otherwise  $A_{ij} = 0$ .  $K_i$  is described as the sum of all valid connected edges of  $V_i$ .

$$K_i = \sum_{j=1}^n A_{ij} \quad (2)$$

Accordingly,  $V_i$  belongs to a community  $S (S \subset G)$ , the degree of  $V_i$  with respect to  $S$  is  $K_i(S) = K_i^{in}(S) + K_i^{out}(S)$ , where  $K_i^{in}(S)$  is the number of edges connecting  $V_i$  to the other vertices in  $S$ , and  $K_i^{out}(S)$  is the number of edges connecting  $V_i$  to the other vertices not in  $S$ . When  $K_i^{in}(S) > K_i^{out}(S), \forall i \in S$ ,  $S$  is seen as a community in a strong sense. Conversely, when  $\sum_{i \in S} K_i^{in}(S) > \sum_{i \in S} K_i^{out}(S)$ ,  $S$  is a weak community. A strong community is more connected within the community than a weak community.

### 2.2. Multi-Objective Optimization

The multi-objective optimization problem returns a set of solutions by balancing a set of conflicting objective functions. In mathematics, taking minimization as an example, a multi-objective optimization problem can be described as follows:

$$\min : F(x) = [f_1(x), f_2(x), \dots, f_m(x)]^T \quad (3)$$

where  $f_i(x)$  is the  $i$ th objective function;  $x$  is the decision variable;  $m$  represents the number of objective functions. Solution  $x_1$  dominates solution  $x_2$ , if the condition is met:

$$\forall f_i(x_1) \leq f_i(x_2) \text{ and } \exists f_i(x_1) < f_i(x_2) (i = 1, 2, \dots, m) \quad (4)$$

Multi-objective optimization returns a set of trade-off non-dominated solutions, rather than an optimal solution. This non-dominated solution set is called the Pareto optimal solution of multi-objective optimization problems. If there is no solution  $x$  dominating  $x^*$ ,

then  $x^*$  is referred to as a Pareto optimal solution or non-dominated solution. A Pareto optimal set or set of non-dominated solutions is defined as:

$$P^* = \{x^* \in X | \nexists x \in X, x \text{ dominate } x^*\} \quad (5)$$

Reference [19] illustrates that the Pareto optimal solution set corresponds to different partitions of a network composed of different numbers of communities. This provides better opportunities for analyzing several communities at different levels. In the multi-objective solution space, the Pareto optimal front (POF) is obtained by mapping these non-dominated solutions [20].

$$POF = \{f_1(x^*), f_1(x^*), \dots, f_m(x^*)^T | x^* \in P^*\} \quad (6)$$

Due to the general applicability of multi-objective optimization, many excellent multi-objective methods have been proposed recently. Leung [34] proposed a collaborative neurodynamic approach for multi-objective optimization that uses weighted Chebyshev to scalarize multiple objectives. In the reconstruction, the multi-projective neural network searches the POF with the help of the PSO algorithm and achieves good performance. Xu [35] designed a fuzzy decision variable framework for large-scale multi-objective optimization to alleviate the problem of too many decision variables hindering the convergence speed of evolutionary algorithms. The framework improves the performance and computational efficiency of the algorithm in large-scale multi-objective optimization through two steps of fuzzy evolution as well as exact evolution; Liu [36] proposed an accelerated evolutionary search strategy for the inefficient decision space of existing multi-objective evolutionary algorithms for dealing with large-scale multi-objective optimization problems. The main idea is to learn a gradient descent direction vector, i.e., the fastest possible convergence direction, for each solution through a specially trained feed-forward neural network to efficiently reconstruct the solution. Experimental results demonstrate that the strategy has obvious advantages in dealing with large-scale multi-objective optimization problems with 1000–10,000 dimensions. These methods perform very well but cannot be applied to the discrete community detection problem.

### 2.3. The Pigeon-Inspired Optimization Algorithm

The pigeon-inspired optimization algorithm (PIO) [37] is a heuristic biomimetic intelligent optimization algorithm proposed by Duan in 2014. This algorithm simulates the flight behavior of pigeons and summarizes two search operations: map and compass operator, as well as landmark operator. In map and compass operators, pigeons move toward the best-positioned pigeon in the group and toward the individual's cognitive direction toward the destination. In the landmark operator, pigeons abandon half of the lost individuals, and the remaining pigeons move toward their destination under the leadership of the elite.

In the PIO algorithm, the position of a virtual pigeon in the solution space is determined by  $x_i (i = 1, 2, \dots, n)$ ;  $v_i (i = 1, 2, \dots, n)$  denotes the flight speed of the pigeon, where  $n$  denotes the number of pigeons. In the early stages of the algorithm, the pigeons rely on the sun as well as the earth's magnetic field for navigation. Each pigeon moves according to the following rules:

$$v_i(t+1) = e^{-Rt} \times v_i(t) + rand \times (x_{gbest}(t) - x_i(t)) \quad (7)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (8)$$

where  $t$  denotes the number of current iterations.  $R$  is a positive real number, which is the map and compass operator.  $R$  normally assumes a value of 0.2.  $rand$  denotes a 0–1 random number that satisfies a normal distribution.  $x_{gbest}$  denotes the global optimal solution.

The map and compass operators attempt to exploit the exploratory power of the pigeon to prevent the algorithm from falling into a local optimal solution. The landmark operator attempts to accelerate the convergence of the algorithm near the optimal solution.

The landmark operator is determined by the position of the center of the current pigeon group:

$$x_{center} = \frac{1}{N(t)} \sum_{i=1}^{N(t)} \frac{W_i}{\sum_{i=1}^{N(t)} W_i} x_j(t) \quad (9)$$

where  $N(t)$  denotes the number of pigeons in the current population;  $W_i$  represents the weight coefficient of the  $i$ th pigeon, calculated according to the following equation:

$$W_i = \frac{1}{f(x_i(t)) + \varepsilon} \quad (10)$$

$\varepsilon$  is a positive real number. The formula for updating the pigeon position in the landmark operator is as follows:

$$x_i(t+1) = x_i(t) + rand \times (x_{center}(t) - x_j(t)) \quad (11)$$

Pigeons who are not familiar with the surrounding environment will gradually be eliminated by the group, according to their fitness value. The number of pigeons in the population after each iteration elimination is:

$$N(t+1) = \frac{N(t)}{2} \quad (12)$$

The basic process of the PIO algorithm (see in Figure 1) is summarized as follows:

**Step 1:** Initialize the position information  $x$  and velocity information  $v$  of the population, as well as other parameters;

**Step 2:** Calculate the fitness value of each pigeon;

**Step 3:** Select the global optimal solution  $x_{gbest}$ ;

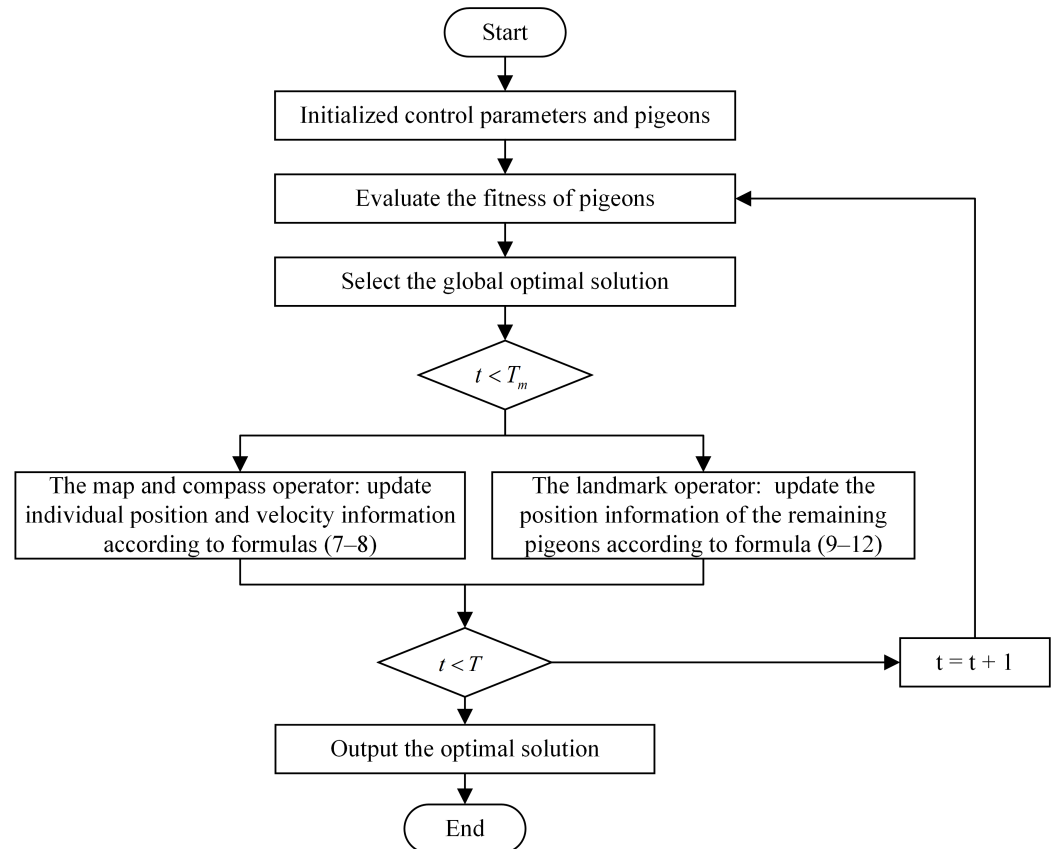
**Step 4:** If the termination condition is not met, skip to step 5, otherwise skip to step 6;

**Step 5:** Update individual position and velocity information according to Formulas (7) and (8);

**Step 6:** Eliminate pigeons and update the position information of the remaining pigeons according to Formula (9);

**Step 7:** If the termination condition is met, output the position information of the pigeon; otherwise,  $t = t + 1$ , jump to step 2.

At this stage, there is less research on community detection based on the PIO algorithms, and only literature [38] has conducted related studies. However, the algorithm exhibits a very disappointing performance. Compared with most multi-objective algorithms for community detection, both the accuracy and stability of community partitioning lag far behind the mainstream community detection algorithms. The superior optimization power of the PIO algorithm is not properly used, which is the starting point of our research in this paper.



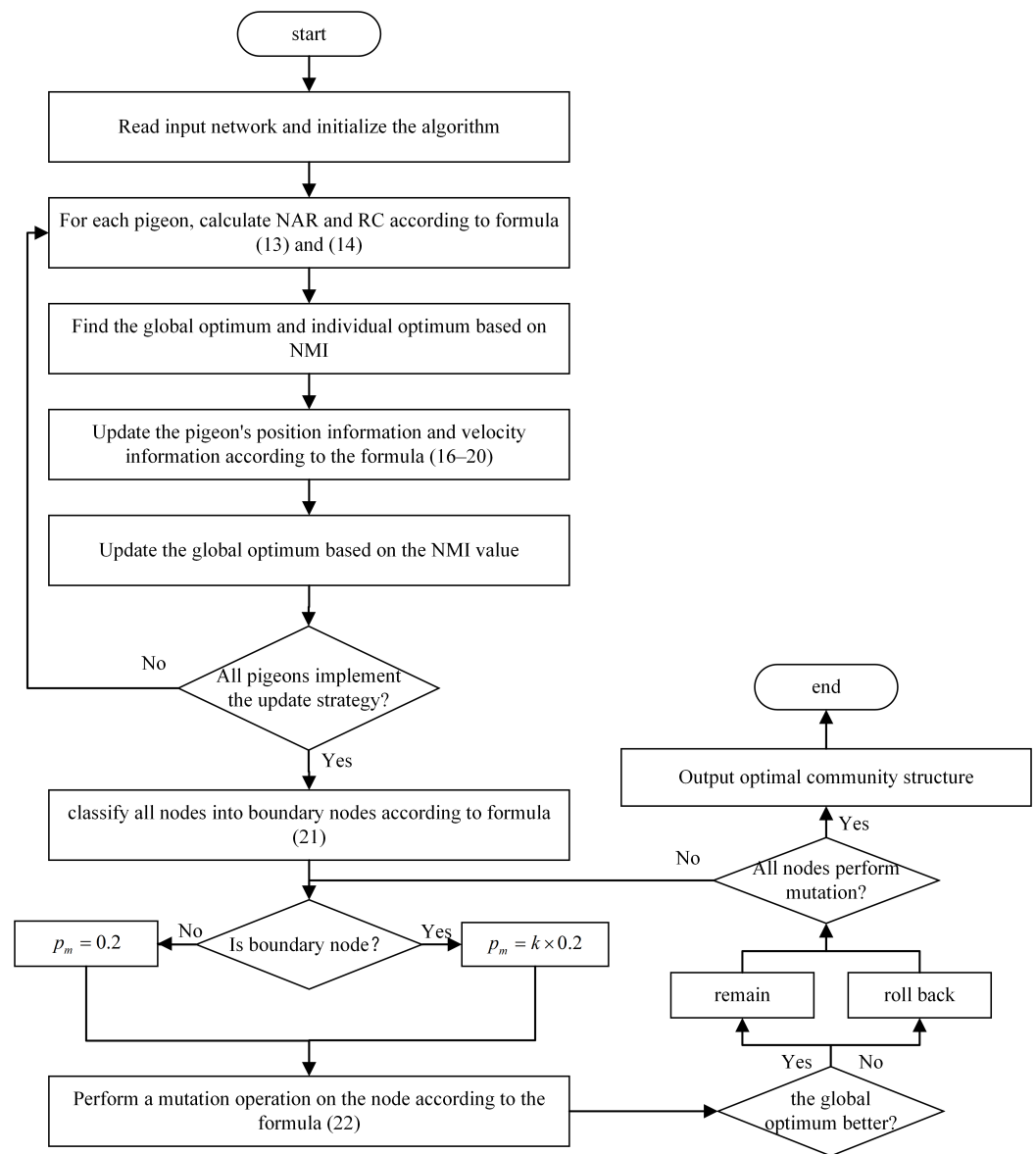
**Figure 1.** The flowchart of the PIO algorithm.

### 3. Proposed Method

Traditional community detection algorithms are mainly based on clustering methods, which have been tested and found to have the disadvantages of both accuracy and complexity. Thanks to the proposal of numerous community structure evaluation functions, intelligent optimization algorithms began to be applied to the field of community detection. Since single-objective optimization algorithms suffer from resolution limitations when optimizing modularity, multi-objective optimization was used. The PIO algorithm is one of the intelligent optimization algorithms with the advantage of high search capability. In this section, the proposed multi-objective pigeon-inspired optimization community detection method called MOPIO-Net is described in detail. The framework of the MOPIO-Net algorithm can be explained in three main steps, including initialization, search, and mutation. During the initialization stage, a specific representation is used to construct the solution, which illustrates a community structure of a network, to clearly and easily display and update the community structure. Thereupon, using this representation, the solutions are initialized by the PGLG method [39]. Then, for each pigeon, two objective functions, including the Negative Ratio Association (NRA) [40] and Ratio Cut (RC) [41] are calculated. In the search phase, inspired by the search strategy in the PIO algorithm, we developed a discretization map and compass operator search process. We try to obtain the local optimal and global optimal solutions by computing the Normalized Mutual Information (NMI) value of each pigeon. In the mutation phase, like the landmark operator in the PIO algorithm, it moves toward the best community structure led by the globally optimal pigeon. This is reflected in the genetic learning of each pigeon with the best global individual. If the community labels on the same gene locus are inconsistent, the mutation will be carried out based on the neighbor's community label. Finally, considering that suboptimal community partitioning is often caused by misclassification of those nodes that are at the community boundaries, we performed a realignment strategy for these boundary nodes



in anticipation of reducing misclassification. The flowchart of the proposed method is illustrated in Figure 2 and additional details are described in the following subsections.



**Figure 2.** The flowchart of the MOPIO-Net algorithm.

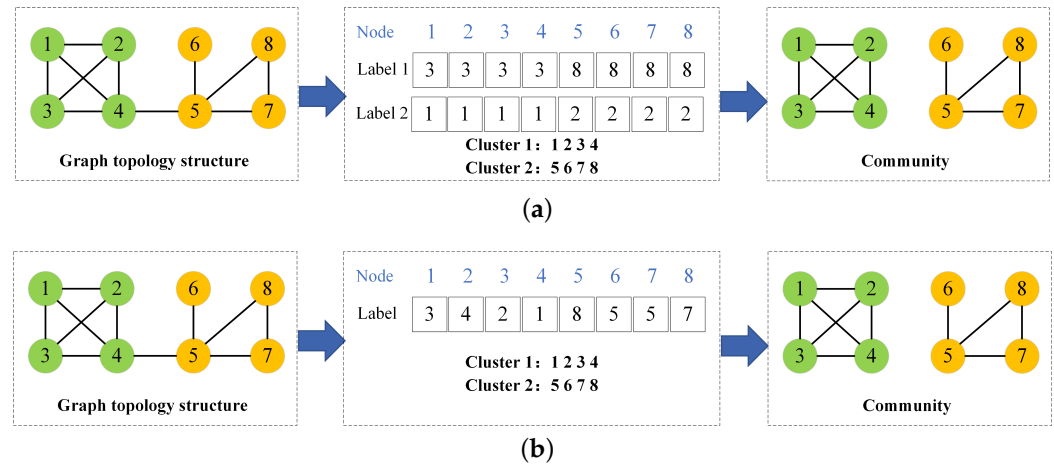
### 3.1. Solution Representation and Initialization

A complex network is essentially a graph structure, and mining its community structure based on intelligent optimization algorithms requires a reasonable representation. To accommodate the discrete optimization problem, the position and velocity of the pigeon swarm are redefined.

#### 3.1.1. Location Representation

Label-based representation and locus-based adjacency representation [42] are two common encoding methods. Both two methods consider each solution as a combination of genes, each of which belongs to a node in the graph. Each gene locus in the locus-based adjacency method is randomly linked to a neighbor node, where  $Gene_i = j$  denotes the existence of linking edges between  $node_i$  and  $node_j$ . This method can automatically obtain the number of communities by decoding, but frequent encoding and decoding operations need to be performed. In the label-based representation method, each gene locus

is generated by tag propagation.  $Gene_i = j$  denotes the  $node_i$  belonging to the  $Community_i$ . If  $Gene_i = Gene_k$ , it means  $node_i$  and  $node_k$  are members of the same community. However, the label-based method has the drawbacks of redundant representation and blind search space. The two representations are shown in Figure 3.



**Figure 3.** Two methods of representation. (a) The label-based representation. (b) The locus-based adjacency representation.

From Figure 3a, although the representations of label 1 and label 2 are different, they represent the same community structure. There are problems with expanding search space, repeating searches, and damaging solution quality when searching. Therefore, we perform redundant operations on the solution based on label-based representation. We obtain the number of communities represented by the current individual based on the community coding at the individual's locus and recode the individual based on the number of communities. As shown in Figure 3a, the label1 position codes only have 3 and 8, thus indicating two communities. We force the community coded in front of the gene position in the individual coding to be 0, 1 is added to the subsequent community coding, and finally, label1 is recoded as  $label1 = [0, 0, 0, 0, 1, 1, 1, 1]$ . The specific algorithm process is shown in Algorithm 1.

---

#### Algorithm 1 Location Representation

---

**begin**

- 1: **for** each  $X_i$  of solutions **do**
  - 2:   Count the number of clusters  $C_{num}$  in the network
  - 3:   **if**  $Max_{label} > C_{num}$  **then**
  - 4:     **for** each label in  $X_i$  **do**
  - 5:       Renumber according to the principle of smaller nodes and smaller numbers
  - 6:     **end for**
  - 7:   **end if**
  - 8: **end for**
- 

#### 3.1.2. Velocity Representation

Velocity guides the flight of pigeons, and a suitable velocity determines whether the pigeons can reach their destination and how fast they can arrive. Excessive velocity can cause pigeons to fly over their destination, while conversely, it can lead to a decrease in the range of the pigeon's activity. The velocity is discretized and expressed as  $V_i = \{v_1, v_2, \dots, v_n\}$ . If  $v_i = 1$ , then it means the label of  $x_i$  in the corresponding position will change; otherwise, the element  $x_i$  remains unchanged.



### 3.2. Fitness Computation

The choice of fitness function is the key to improving the quality of solution optimization, whether in multi-objective optimization problems or in community detection problems. We optimize the objective functions with RA and RC. RA represents the average number of edges that exist between nodes in all communities. The value of RA is inversely proportional to the number of communities in the network. The larger the value of RA, the smaller the number of large communities with a high density of internal connections into which the network will be divided. The average values of connections between nodes within a community and other communities represented by RC are summed. The value of RC is proportional to the number of communities in the network. The smaller the value of RC, the sparser the edges connected between communities, and the greater amount of nodes within the community. This will divide the entire network into a smaller number of community structures with high internal connection density. We chose these two metric functions because, as we mentioned in the previous section, the tighter the intra-community connections and the sparser the inter-community connections, the clearer the community structure and the higher the algorithm recognition accuracy. From Equations (13) and (14), we can observe that the RC denotes the ratio of the number of inter-community edges to the number of communities. The RA denotes the ratio of the number of intra-community edges to the number of communities. In order to obtain a clearer community structure, when randomly grouping nodes, it is desired that the number of intra-community edges is as high as possible (RA) and the number of inter-community edges is as low as possible (RC). To formulate the problem as a minimum optimization problem, we take the opposite of the objective function RA, called the negative ratio association (NRA). Both objective functions are minimized simultaneously, allowing the community partitioning results to be explored toward the community structure we expect to obtain (internally tight, externally sparse).

Assume an undirected acyclic graph  $G = (V, E)$  contains  $|V|$  nodes and  $|E|$  edges. The corresponding adjacency matrix is  $A$ . A community structure  $C = \{C_1, C_2, \dots, C_m\}$  denotes the division of the graph  $G$  into  $m$  communities. In the non-overlapping community detection study,  $L(C_1, C_2)$  defines the number of edge connections that exist between two communities. The two objective functions are formulated as follows:

$$RC = \sum_{i=1}^m \frac{L(C_i, \bar{C}_i)}{|C_i|} \quad (13)$$

$$NRA = -RA = -\sum_{i=1}^m \frac{L(C_i, C_i)}{|C_i|} \quad (14)$$

$$L(C_i, C_j) = \sum_{i \in C_i, j \in C_j} A_{ij} \quad (15)$$

### 3.3. Search Strategy

In the PIO algorithm, the pigeons follow the global optimal solution at the map and compass operator and the central solution at the landmark operator. In the community detection problem, we utilize the mutation operation instead of the original search strategy for the second stage.

In the discrete process, the update rule for redefining the pigeon's velocity is:

$$V_i^{t+1} = \text{sig}(e^{-Rt} V_i^t + \text{rand}(X_{g\text{best}} \oplus X_i^t)) \quad (16)$$

where  $\oplus$  denotes the XOR operator. The role of the  $\text{sig}()$  function is to map the velocity into  $[0, 1]$  space, and  $y = \text{sig}(V)$  is defined as:

$$\begin{cases} y_i = 0, & \text{if } \text{rand}(0, 1) \geq \text{sigmoid}(v_i) \\ y_i = 1, & \text{if } \text{rand}(0, 1) < \text{sigmoid}(v_i) \end{cases} \quad (17)$$

where the sigmoid function rule is:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (18)$$

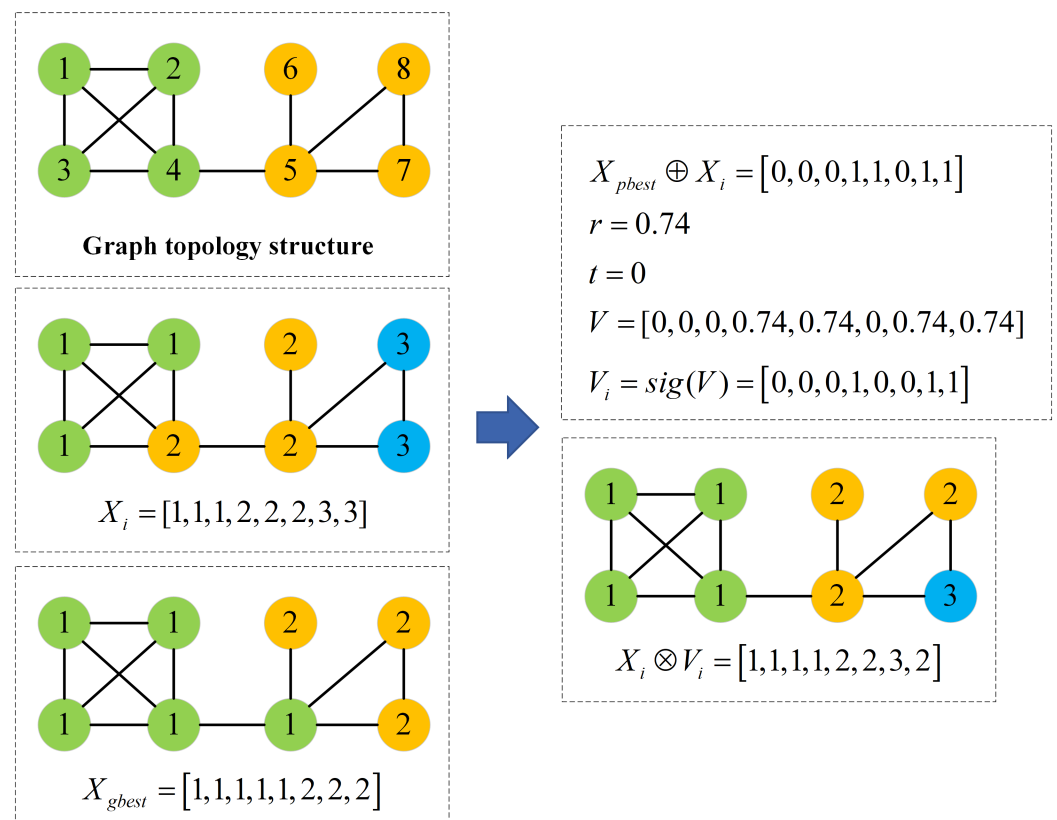
Based on the redefined velocity update rule, we now represent the pigeon's position update rule in the following discrete form:

$$X_i^{t+1} = X_i^t \otimes V_i^t \quad (19)$$

The above equation indicates that during the  $t$ th iteration,  $X_i$  generates new position information  $X_i^{t+1} = \{x_{i1}^{t+1}, x_{i2}^{t+1}, \dots, x_{in}^{t+1}\}$  guided by the velocity  $V_i$ . The specific computation rules for the  $\otimes$  operator are:

$$\begin{cases} x_{ij}^{t+1} = x_{ij}^t, & \text{if } v_{ij}^t = 0 \\ x_{ij}^{t+1} = \text{Max}N_j, & \text{if } v_{ij}^t = 1 \end{cases} \quad (20)$$

Among them,  $\text{Max}N_j$  is a positive integer that represents the label with the highest frequency in the neighbor set of  $\text{node}_j$ . We choose this method to update location information because the more neighbors a node joins in the community, the closer the community structure is internally and the sparser it is externally. This is exactly the community structure we expect to detect. The schematic diagram of the overall search process in the first stage can be found in Figure 4.



**Figure 4.** The first phase of the search update strategy.

Boundary nodes connect multiple communities, and their neighbors belong to different communities. Compared to non-boundary nodes, boundary nodes are more prone to misclassification. The misclassification of boundary nodes is one of the main factors leading to poor community structure. Therefore, to improve the quality of the partitioning results, we conducted different strategy mutation operations on boundary nodes and non-

boundary nodes. Set the mutation probability to  $P_m$ . If the node belongs to a boundary node, the probability of mutation is increased accordingly to the number of types of community labels that the neighbors belong to. The specific mutation rules are as follows, where  $k$  is the count of the different communities to which  $node_i$ 's neighbor nodes belong. The second phase of the search update strategy can be summarized as shown in Figure 5.

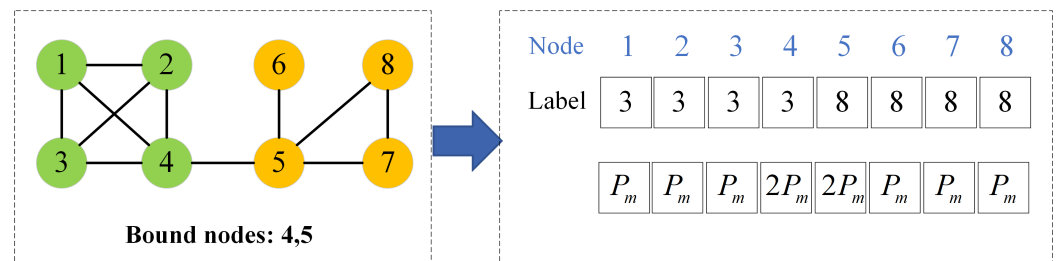


Figure 5. Schematic diagram of boundary node strategy.

In the second search phase, the algorithm pseudo-code is shown in Algorithm 2.

---

#### Algorithm 2 Mutation

---

**begin**

```

1: for each node in  $X_i$  do
2:   for each neighbor of node do
3:     count the number of different labels ( $k_{node}$ )
4:   end for
5:   if  $rand < k_{node} \times P_m$  then
6:      $label_{node} = MaxN_{node}$ 
7:   end if
8:   if  $X_{new}$  dominate  $X_i$  then
9:      $X_i = X_{new}$ 
10:  else
11:    rollback
12:  end if
13: end for

```

---

The MOPIO-Net algorithm has three main processes: initialization, search and mutation. The first process complexity is analyzed according to Algorithm 1 as  $O(popsiz * n)$ ; the second process complexity is mainly calculated by the fitness as well as the position update, and the complexity is  $O(2 * popsize)$ . The complexity of the third process of mutation is analyzed according to Algorithm 2 as  $O(popsiz * n)$ . Therefore, the complexity of MOPIO-Net is  $O(Gen * popsize * n)$ , where  $popsiz$  denotes the population size,  $n$  denotes the number of nodes in the network, and  $Gen$  denotes the number of iterations.

#### 4. Experiment

In this section, we evaluate the performance of the proposed MOPIO-Net algorithm against 10 outstanding community detection methods on an extended Girvan Network benchmark. The algorithmic community detection results were then compared across several real-world networks, such as the Zachary Karate Club Network, the American College Football Network, and the Dolphin Network. The baseline community detection methods that we compared are shown in Table 1 below.

**Table 1.** A brief introduction to eleven baseline algorithms, and the corresponding literature.

Method	Description	In Ref
GA-Net	Single objective optimization method <b>Fitness function:</b> Modularity	[14]
BGLL	Single objective optimization method <b>Fitness function:</b> Modularity	[43]
Meme-Net	Single objective optimization method <b>Fitness function:</b> module density mass function.	[16]
MOGA-Net	Multi-objective optimization method <b>Fitness function:</b> Community score, community fitness	[19]
MOEA/D-Net	Multi-objective optimization method <b>Fitness function:</b> RC, NRA	[44]
MOPSO-Net	Multi-objective optimization method <b>Fitness function:</b> Kernel K-Means, RC	[20]
MODPSO	Multi-objective optimization method <b>Fitness function:</b> Kernel K-Means, RC	[39]
MOPIO	Multi-objective optimization method <b>Fitness function:</b> NRA, RC	[38]
MOCD-ACO	Multi-objective optimization method <b>Fitness function:</b> NRA, RC	[45]
MODCRO	Multi-objective optimization method <b>Fitness function:</b> Kernel K-Means, RC	[25]

#### 4.1. Parameter Setting

The MOPIO-Net algorithm was implemented on PyCharm 2022. The algorithm relies on a computer with an Intel *Core<sup>TM</sup>* i5 CPU 2.67 GHz and 16 GB (14 GB usable) of memory configuration. The parameters of all the above algorithms were set according to the corresponding papers, the population size was set to 100, and the maximum number of iterations was set to 100; the map and compass operators' iteration was set to 80; the landmark operators' iteration was set to 20 and the mutation operator  $P_m = 0.2$ .

#### 4.2. Evaluation Metric

To evaluate the performance of the MOPIO-Net algorithm with other baseline algorithms in the community detection problem, we chose NMI as the evaluation metric. The NMI measures the similarity between the real community structure of the network and the community structure identified by the algorithm. The more similar the community structure identified by the algorithm is to the real community structure, the closer the value of NMI is to 1; on the contrary, the value of NMI is close to 0. The value of  $NMI \in [0, 1]$ . Suppose, the real community structure partition is  $D = \{D_1, D_2, \dots, D_q\}$  and the community structure partition identified by the algorithm is  $E = \{E_1, E_2, \dots, E_p\}$ , where  $q$  and  $p$  denote the number of communities in the real partition  $D$  and the algorithm's partition  $E$ , respectively. By introducing the confusion matrix  $C$  and calculating the similarity of the two partitions, the NMI can be defined as:

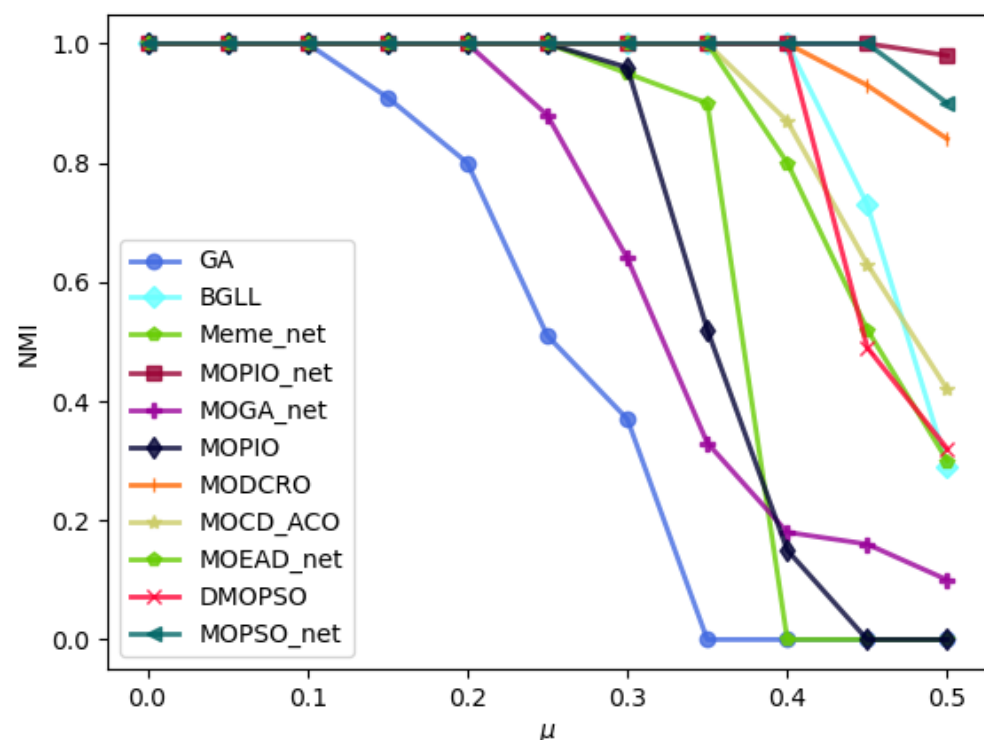
$$NMI = \frac{-2 \sum_{i=1}^{C_D} \sum_{j=1}^{C_E} C_{ij} \log(C_{ij}/C_{i.}C_{.j})}{\sum_{i=1}^{C_D} C_{i.} \log(C_{i.}/N) + \sum_{j=1}^{C_E} C_{.j} \log(C_{.j}/N)} \quad (21)$$

where  $C_{D(E)}$  denotes the number of communities in partition  $D(E)$ ,  $C_{i.(.j)}$  denotes the sum of the elements in row  $i$  (column  $j$ ) of the confusion matrix, and  $N$  denotes the number of nodes in the network.

#### 4.3. Experimental Results of the GN Extended Benchmark Network

The nodes in the GN extended benchmark network form four communities, each containing 32 nodes, totaling 128 nodes. The average degree of nodes in the network is 16. The GN extended benchmark network is an overt synthetic network with the complexity controlled by the parameter  $\mu$ . Each node within the same community is connected to other nodes in the community with probability  $1 - \mu$ . We expect tightly connected nodes within a community and sparse connections between communities ( $\mu \rightarrow 0$ ), such that the network has low-performance requirements for the algorithm. Increasing the value of  $\mu$ , the more ambiguous the network structure is, which is the opposite of our desired network topology. When  $\mu = 0.5$ , the proportion of edge connections within a community is the same as the proportion of edge connections between communities, which is a great challenge for the community detection algorithm. We performed 11 sets of comparison experiments of the GN extended benchmark networks, corresponding to networks generated with parameters  $\mu$  from 0 to 0.5 at 0.05 intervals.

Figure 6 shows the average NMI values obtained for 10 independent runs of each algorithm on the GN extended network. All algorithms detect the true community structure when  $\mu$  takes the values of 0, 0.05, and 0.1. The GA-Net algorithm is unable to identify the true community structure at  $\mu = 0.15$ , which is far below the maximum upper limit of  $\mu$  when other algorithms can identify the true community. As the parameter  $\mu$  increases, our proposed MOPIO-Net algorithm exhibits the best performance results. When  $\mu = 0.5$ , the community structure identified by MOPIO-Net has only a few node misclassifications from the real situation, and the NMI value is close to 1. In addition, the MOPSO-Net and the MODCRO exhibit accuracy rates second only to the MOPIO-Net algorithm. The MOPIO-Net algorithm can identify the real community structure more accurately when the network community structure is ambiguous.

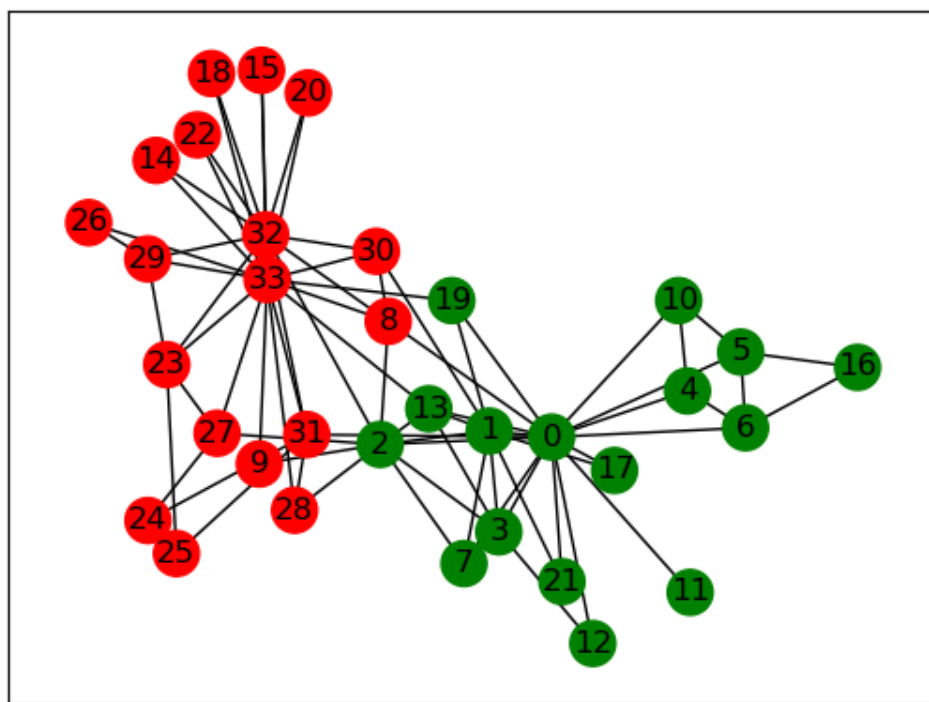


**Figure 6.** The value of NMI obtained by 10 baseline algorithms and the MOPIO-Net algorithm on the extension of the classical GN extended benchmark network.

#### 4.4. Experimental Results on the Real Networks

Three classical real networks, the Karate Network [45], the Dolphin Network [46], and the American College Football Network [11], were chosen to test the performance of all algorithms.

The Karate Network was established by Zachary in 1977. The karate network mainly recorded the social network of 34 karate members over a 2-year period due to disagreements between coaches and administrators. The real karate network community structure is shown in Figure 7.



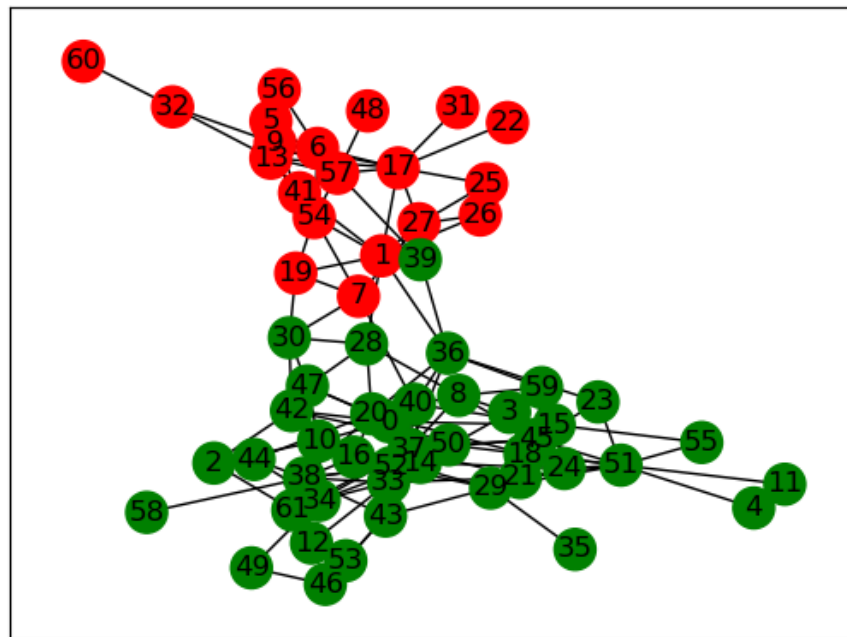
**Figure 7.** The Zachary's karate club network.

The Bottlenose Dolphin Network is a network established by David Lusseau in 2003. The Bottlenose Dolphin Network focuses on how often the 62 bottlenose dolphins living in New Zealand's Doubtful Sound are counted by humans, with higher count frequency indicating a more likely connection between two bottlenose dolphins. The real partition of the bottlenose dolphin network is shown in Figure 8.

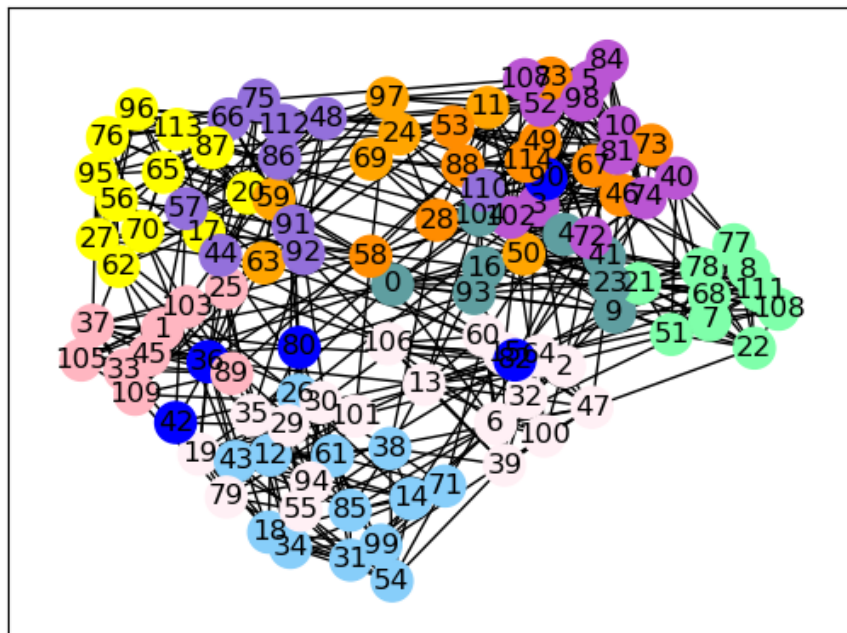
The American College Football Network was statistically completed by Girvan and Newman in 2000. The American College Football Network simulates the regular season schedule of teams in the NCAA Division I football league for the 2000 season. The real community structure of the American College Football Network is shown in Figure 9. The details of each network are shown in Table 2.

**Table 2.** The number of nodes, the number of edges and the number of communities in the real network.

Networks	Number of Nodes	Number of Edges	Number of Communities
Zackary's karate club	34	78	2
Dolphin network	62	159	2
American College Football Network	115	613	12



**Figure 8.** The Bottlenose Dolphins network.

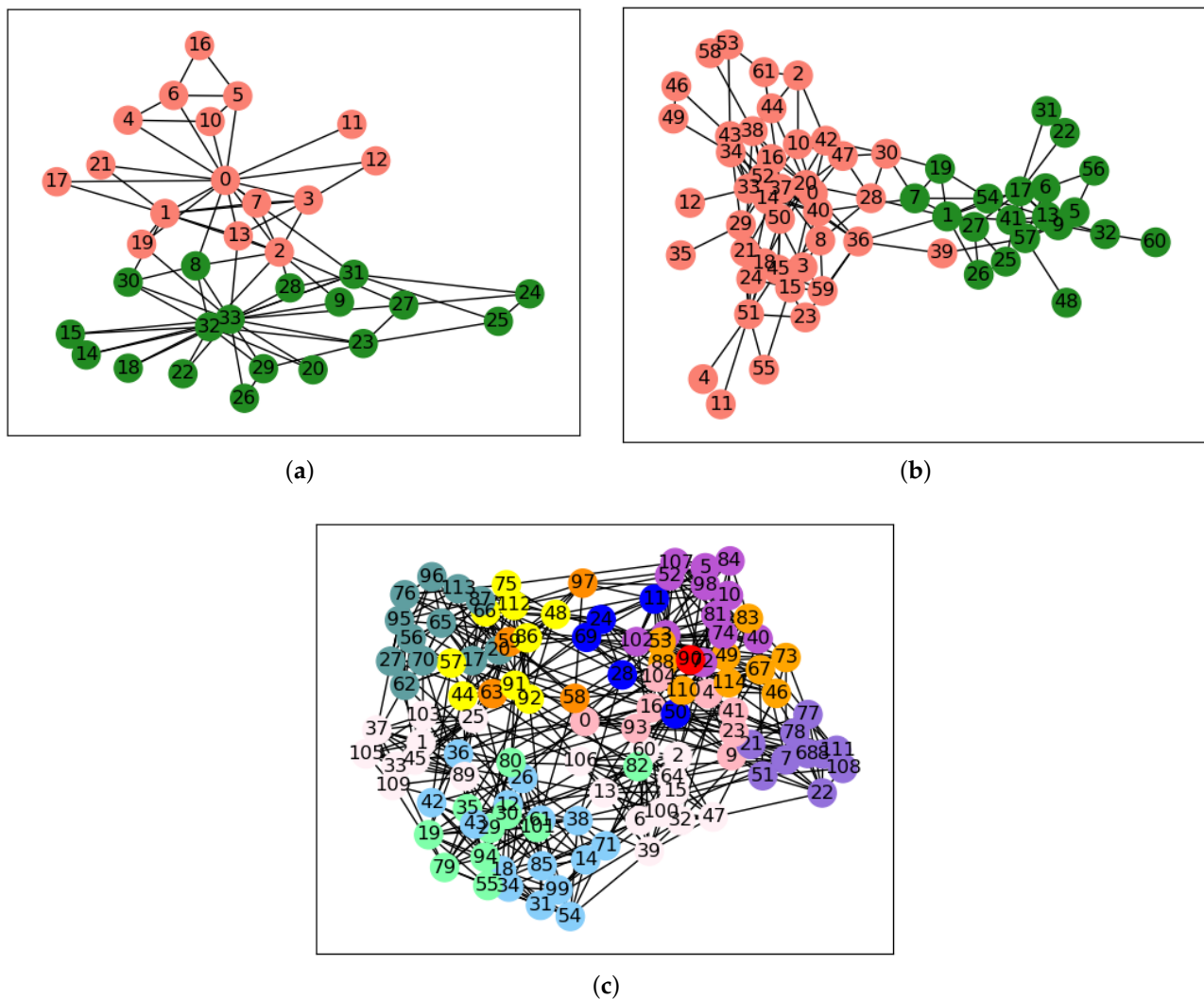


**Figure 9.** The American College Football network.

Table 3 shows the NMI values of the detection results of all algorithms under the three real networks. The final partitioning results of MOPIO-Net under three real networks are shown in the following Figure 10. The karate network and the Dolphin network were consistently consistent with the true results in 20 independent runs. In the American College Football network, we have just one node, No.90, with the wrong partition. At this point, the NMI value of the partitioning result is 0.932. The recognition accuracy exhibited by the single-objective optimization algorithm is clearly unsatisfactory. The GA-Net algorithm and BGLL algorithm have lower NMI values, where BGLL is mainly oriented to large-scale network recognition and does perform well in US college football network recognition, and has higher NMI values compared to several multi-objective optimization algorithms such as the Meme-Net algorithm, MOGA-Net algorithm and MOPIO algo-



rithm. The multi-objective optimization algorithm has more stable recognition and a higher  $NMI_{avg}$  value when dealing with the community detection problem. Using a set of conflicting optimization objective functions does improve the average accuracy of recognition. Our proposed algorithm MOPIO-Net shows the best partitioning in both the Karate network, the Dolphin network and the American College Football network with high values of  $NMI_{avg}$ . The higher accuracy compared to other multi-objective optimization algorithms (MOGA-Net, MOPIO and Meme-Net) is because our algorithm includes not only crossover and mutation but also the search strategy of the underlying algorithm. The reason for the higher accuracy compared to multi-objective optimization algorithms such as MOPSO-Net and MODPSO is our treatment of boundary nodes, which reduces the interference factor of identification from the error source. This indicates that the enhancement of community detection partitioning accuracy by crossover and mutation alone is limited. In terms of stability, MOPIO-Net is analyzed from the values in the table and performs better on the basis of guaranteed accuracy. On the one hand, MOPIO-Net stability is affected by the randomized algorithm itself. On the other hand, it is also affected by the update strategy that is affected by the update strategy. Each node may be randomly selected when approaching its neighbors, which affects the stability of the algorithm.



**Figure 10.** (a) The detected clusters of the best result of the MOPIO-Net on Zachary's karate network. (b) The detected communities of the best result of the MOPIO-Net on the Bottlenecked Dolphin network. (c) The detected community structure of the MOPIO-Net on the American Football network.

**Table 3.** Results obtained by the eleven methods on three real networks.

Method	Karate			Dolphin			Football		
	$NMI_{\max}$	$NMI_{avg}$	$NMI_{std}$	$NMI_{\max}$	$NMI_{avg}$	$NMI_{std}$	$NMI_{\max}$	$NMI_{avg}$	$NMI_{std}$
GA-Net	1	0.6654	0.3221	0.6267	0.6264	0.0001	0.9104	0.8977	0.0253
BGLL	1	0.7076	0.2912	0.6956	0.5144	0.1451	0.8358	0.8358	0
Meme-Net	1	0.8644	0.1221	1	0.7889	0.3103	0.8616	0.7669	0.0897
MOGA-Net	1	1	0	1	0.9389	0.0057	0.8045	0.7950	0.0015
MOEA/D-Net	1	1	0	1	1	0	0.9296	0.9294	0.0001
MOPSO-Net	1	1	0	1	1	0	0.9325	0.9316	0.0004
MODPSO	1	1	0	1	1	0	0.9298	0.9278	0.0008
MOPIO	1	0.860	0.2242	1	0.8022	0.2442	0.8160	0.7542	0.0606
MOCD-ACO	1	1	0	1	1	0	0.9374	0.9286	0.0117
MODCRO	1	0.9673	0.0197	1	0.9495	0.0377	0.9000	0.8674	0.0412
MOPIO-Net	<b>1</b>	<b>1</b>	0	<b>1</b>	<b>1</b>	0	<b>0.9423</b>	<b>0.9336</b>	0.0091

Bold represents the best results by default in the experimental data.

## 5. Conclusions

In this paper, a novel multi-objective community detection algorithm based on a discrete PIO named MOPIO-Net has been proposed. The proposed method uses a multi-objective optimization strategy to solve the community detection problem. Our proposed method minimizes the set of conflicting objective functions, NRA and RC, to obtain a partition structure with tight intra-community connectivity and sparse inter-community connectivity. We changed the movement strategy of the pigeon in the PIO algorithm. In the new strategy, a similar crossover operation is performed by the pigeon to move closer to the optimal solution. For the community detection misclassification problem due to boundary nodes, we implemented different strategies for the community classification of boundary nodes. To verify the performance of the MOPIO-Net algorithm, a synthetic network and three real networks were tested. The results were compared with 11 excellent community detection algorithms. The experimental results show that MOPIO-Net detects partitions closer to the real community structure under all networks. It is verified that our discretization strategy is feasible, the algorithm avoids the resolution limitation problem, and the proposed boundary node variation strategy further improves the recognition accuracy.

The work in this paper validates the effectiveness of the MOPIO-Net algorithm in static network community detection. We hope to further explore the possibilities of MOPIO-Net in overlapping networks and dynamic networks in the future. We also consider how MOPIO-Net should deal with special networks such as signed networks and weighted networks.

**Author Contributions:** Writing—original draft, L.Y.; validation, X.G.; writing—review and editing, D.Z.; methodology, J.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Provincial Frontier leading technology basic research major project fund under grant number BK20232028.

**Data Availability Statement:** <https://websites.umich.edu/~mejn/netdata/> (accessed on 21 September 2023).

**Conflicts of Interest:** Authors Xiaodan Guo and Dongdong Zhou were employed by Northern Information Control Research Institute Group Co. The authors declare that they have no competing financial interests or personal relationships that may have influenced the work reported in this study.

## References

1. Bennett, L.; Kittas, A.; Muirhead, G.; Papageorgiou, L.G.; Tsoka, S. Detection of Composite Communities in Multiplex Biological Networks. *Sci. Rep.* **2015**, *5*, 10345. [CrossRef]
2. Tamura, K.; Kobayashi, Y.; Ihara, Y. Evolution of individual versus social learning on social networks. *J. R. Soc. Interface* **2015**, *12*, 20141285. [CrossRef]
3. Harakawa, R.; Ogawa, T.; Haseyama, M. Accurate and Efficient Extraction of Hierarchical Structure of Web Communities for Web Video Retrieval. *ITE Trans. Media Technol. Appl.* **2015**, *4*, 49–59.
4. Khanfor, A.; Ghazzai, H.; Yang, Y.; Massoud, Y. Application of Community Detection Algorithms on Social Internet-of-things Networks. In Proceedings of the 31st International Conference on Microelectronics (IEEE ICM 2019), Cairo, Egypt, 15–18 December 2019; pp. 94–97.
5. Rostami, M.; Berahmand, K.; Forouzandeh, S. A novel community detection based genetic algorithm for feature selection. *J. Big Data* **2021**, *8*, 2. [CrossRef]
6. Moradi, P.; Ahmadian, S.; Akhlaghian, F. An effective trust-based recommendation method using a novel graph clustering algorithm. *Phys. A-Stat. Mech. Its Appl.* **2015**, *436*, 462–481. [CrossRef]
7. Rezaeimehr, F.; Moradi, P.; Ahmadian, S.; Qader, N.N.; Jalili, M. TCARS: Time- and Community-Aware Recommendation System. *Future Gener. Comput. Syst. Int. J. Escience* **2018**, *78*, 419–429. [CrossRef]
8. Wang, Z.; Wu, Y.; Li, Q.; Jin, F.; Xiong, W. Link prediction based on hyperbolic mapping with community structure for complex networks. *Phys. A-Stat. Mech. Its Appl.* **2016**, *450*, 609–623. [CrossRef]
9. Deng, X.; Wen, Y.; Chen, Y. Highly efficient epidemic spreading model based LPA threshold community detection method. *Neurocomputing* **2016**, *210*, 3–12. [CrossRef]
10. Wang, S.; Gong, M.; Liu, W.; Wu, Y. Preventing epidemic spreading in networks by community detection and memetic algorithm. *Appl. Soft Comput.* **2020**, *89*, 106118. [CrossRef]

11. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [[CrossRef](#)]
12. Newman, M. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *69*, 066133. [[CrossRef](#)]
13. Lancichinetti, A.; Fortunato, S.; Kertesz, J. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* **2009**, *11*, 033015. [[CrossRef](#)]
14. Pizzuti, C. GA-Net: A Genetic Algorithm for Community Detection in Social Networks. In *Parallel Problem Solving from Nature—PPSN X, Proceedings of the 10th International Conference on Parallel Problem Solving from Nature, Dortmund, Germany, 13–17 September 2008*; Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N., Eds.; Lecture Notes in Computer Science Series; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5199, pp. 1081–1090.
15. Li, J.; Song, Y. Community detection in complex networks using extended compact genetic algorithm. *Soft Comput.* **2013**, *17*, 925–937. [[CrossRef](#)]
16. Gong, M.; Fu, B.; Jiao, L.; Du, H. Memetic algorithm for community detection in networks. *Phys. Rev. E* **2011**, *84*, 056101. [[CrossRef](#)]
17. Fortunato, S.; Barthelemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 36–41. [[CrossRef](#)]
18. Tian, Y.; Yang, S.; Zhang, X. An Evolutionary Multiobjective Optimization Based Fuzzy Method for Overlapping Community Detection. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 2841–2855. [[CrossRef](#)]
19. Pizzuti, C. A Multiobjective Genetic Algorithm to Find Communities in Complex Networks. *IEEE Trans. Evol. Comput.* **2012**, *16*, 418–430. [[CrossRef](#)]
20. Rahimi, S.; Abdollahpouri, A.; Moradi, P. A multi-objective particle swarm optimization algorithm for community detection in complex networks. *Swarm Evol. Comput.* **2018**, *39*, 297–309. [[CrossRef](#)]
21. Messaoudi, I.; Kamel, N. A multi-objective bat algorithm for community detection on dynamic social networks. *Appl. Intell.* **2019**, *49*, 2119–2136. [[CrossRef](#)]
22. Li, Q.; Cao, Z.; Ding, W.; Li, Q. A multi-objective adaptive evolutionary algorithm to extract communities in networks. *Swarm Evol. Comput.* **2020**, *52*, 100629. [[CrossRef](#)]
23. Chen, D.; Zou, F.; Lu, R.; Yu, L.; Li, Z.; Wang, J. Multi-objective optimization of community detection using discrete teaching-learning-based optimization with decomposition. *Inf. Sci.* **2016**, *369*, 402–418. [[CrossRef](#)]
24. Ji, P.; Zhang, S.; Zhou, Z. A decomposition-based ant colony optimization algorithm for the multi-objective community detection. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 173–188. [[CrossRef](#)]
25. Li, H.; Gan, W. A Decomposition-Based Multiobjective Chemical Reaction Optimization Algorithm for Community Detection in Complex Networks. *Int. J. Comput. Intell. Syst.* **2020**, *13*, 524–537. [[CrossRef](#)]
26. Perez-Pelo, S.; Sanchez-Oro, J.; Gonzalez-Pardo, A.; Duarte, A. A fast variable neighborhood search approach for multi-objective community detection. *Appl. Soft Comput.* **2021**, *112*, 107838. [[CrossRef](#)]
27. Ma, H.; Yang, H.; Zhou, K.; Zhang, L.; Zhang, X. A local-to-global scheme-based multi-objective evolutionary algorithm for overlapping community detection on large-scale complex networks. *Neural Comput. Appl.* **2021**, *33*, 5135–5149. [[CrossRef](#)]
28. Jokar, E.; Mosleh, M.; Kheyrandish, M. GWBM: An algorithm based on grey wolf optimization and balanced modularity for community discovery in social networks. *J. Supercomput.* **2022**, *78*, 7354–7377. [[CrossRef](#)]
29. 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*, 437–449. [[CrossRef](#)]
30. Pizzuti, C.; Socievole, A. Multiobjective Optimization and Local Merge for Clustering Attributed Graphs. *IEEE Trans. Cybern.* **2020**, *50*, 4997–5009. [[CrossRef](#)]
31. Besharatnia, F.; Talebpour, A.; Aliakbary, S. An Improved Grey Wolves Optimization Algorithm for Dynamic Community Detection and Data Clustering. *Appl. Artif. Intell.* **2022**, *36*, 2012000. [[CrossRef](#)]
32. Fortunato, S. Community detection in graphs. *Phys. Rep.* **2010**, *486*, 75–174. [[CrossRef](#)]
33. 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](#)] [[PubMed](#)]
34. Leung, M.F.; Wang, J. A Collaborative Neurodynamic Approach to Multiobjective Optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5738–5748. [[CrossRef](#)] [[PubMed](#)]
35. Yang, X.; Zou, J.; Yang, S.; Zheng, J.; Liu, Y. A Fuzzy Decision Variables Framework for Large-Scale Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2023**, *27*, 445–459. [[CrossRef](#)]
36. Liu, S.; Li, J.; Lin, Q.; Tian, Y.; Tan, K.C. Learning to Accelerate Evolutionary Search for Large-Scale Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2023**, *27*, 67–81. [[CrossRef](#)]
37. 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](#)]
38. 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. [[CrossRef](#)]
39. 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](#)]
40. Angelini, L.; Boccaletti, S.; Marinazzo, D.; Pellicoro, M.; Stramaglia, S. Identification of network modules by optimization of ratio association. *Chaos* **2007**, *17*, 023114. [[CrossRef](#)] [[PubMed](#)]

41. Wei, Y.C.; Cheng, C.K. Ratio cut partitioning for hierarchical designs. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **1991**, *10*, 911–921. [[CrossRef](#)]
42. Pizzuti, C. Evolutionary Computation for Community Detection in Networks: A Review. *IEEE Trans. Evol. Comput.* **2018**, *22*, 464–483. [[CrossRef](#)]
43. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.-Theory Exp.* **2008**, *2008*, P10008. [[CrossRef](#)]
44. Gong, M.; Ma, L.; Zhang, Q.; Jiao, L. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Phys. A-Stat. Mech. Its Appl.* **2012**, *391*, 4050–4060. [[CrossRef](#)]
45. Zachary, W.W. An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropol. Res.* **1977**, *33*, 452–473. [[CrossRef](#)]
46. Lusseau, D.; Schneider, K.; Boisseau, O.; Haase, P.A.; Slooten, E.; Dawson, S.M. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **2003**, *54*, 396–405. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.