entropy

MDPI

Article

Detecting Overlapping Communities in Modularity Optimization by Reweighting Vertices

Chen-Kun Tsung [1,*], Hann-Jang Ho [2], Chien-Yu Chen [3], Tien-Wei Chang [3] and Sing-Ling Lee [3]

1 Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan
2 Department of Applied Digital Media, WuFeng University, Chiayi County 62153, Taiwan; hhj@wfu.edu.tw
3 Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 62102, Taiwan; kw.15@hotmail.com (C.-Y.C.); felicitylab@gmail.com (T.-W.C.); singling@ccu.edu.tw (S.-L.L.)
* Correspondence: ckt@ncut.edu.tw; Tel.: +886-4-23924505

Received: 21 June 2020; Accepted: 24 July 2020; Published: 27 July 2020

Abstract: On the purpose of detecting communities, many algorithms have been proposed for the disjointed community sets. The major challenge of detecting communities from the real-world problems is to determine the overlapped communities. The overlapped vertices belong to some communities, so it is difficult to be detected using the modularity maximization approach. The major problem is that the overlapping structure barely be found by maximizing the fuzzy modularity function. In this paper, we firstly introduce a node weight allocation problem to formulate the overlapping property in the community detection. We propose an extension of modularity, which is a better measure for overlapping communities based on reweighting nodes, to design the proposed algorithm. We use the genetic algorithm for solving the node weight allocation problem and detecting the overlapping communities. To fit the properties of various instances, we introduce three refinement strategies to increase the solution quality. In the experiments, the proposed method is applied on both synthetic and real networks, and the results show that the proposed solution can detect the nontrivial valuable overlapping nodes which might be ignored by other algorithms.

Keywords: data mining; community detection; overlapping communities; modularity

1. Introduction

Determining the group with some particular properties helps the analysts to capture the common properties from the members in the community. Many applications could be considered based on the community detection. For example, the precise information delivery, e.g., Google AdWords [1] increases the transaction amounts for sending the advertisement information to the right person. Therefore, detecting communities is a popular research topic [2–8].

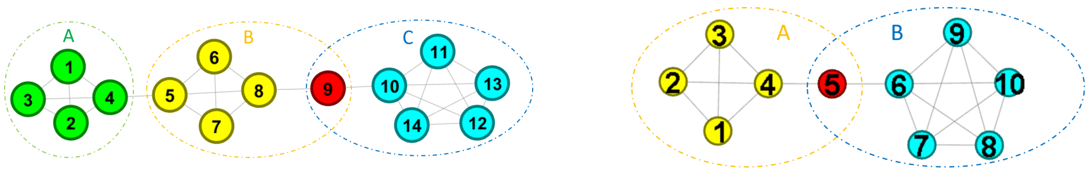
Many results focus on the disjoin community sets that each node belongs to exactly one community [2,3]. However, in the real-world networks, many people may belong to multiple communities, so the communities may overlap with each other. For example, an engineer may belong to many projects in a company. Thus, instead of strict partitions, fuzzy partitions are more appropriate for understanding the network structures [9,10]. Fuzzy partitions allow a node belongs to multiple communities simultaneously. Considering a real-world situation, some staff work together in a building, and the manager would like to track the movement history for each staff [11]. Each one may move to various rooms, and the move purpose comes from the role of each staff. When we treat the purpose of all staff to be the communities, the staff may belong to different communities.

The modularity function proposed by Newman and Girvan [12] is the famous measurement of network partitions to measure the structure of a given network. The modularity function calculates the difference between the number of real intra-community edges and the expected number of edges to identify the qualities of the communities. The partition with larger modularity value has better community structure than those with lower modularity values. Finding the partitions with maximum modularity is a straightforward solution to the community detection. However, the modularity maximization has been proved as an NP-hard problem [13], and finding the partition with maximum modularity is difficult. Therefore, many results are proposed to calculate the near optimal solutions, such as the random walk processes [14], the structural clustering [15], and the polynomial-time approximation algorithms [16].

On the other hand, besides the computation complexity, the modularity maximization has two problems in detecting communities:

Resolution limits Fortunato et al. introduced that small communities cannot be detected in large networks [17,18]. Since the null model of modularity provides the global connectivity, the expected number of edges between two small groups in a large network might be very small. Eventually, the two small groups will be treated as one community. Many approaches are proposed for solving resolution limits to provide high solution qualities, such as greedy algorithms [19,20], spectral algorithms [21–23], simulating annealing algorithms [24] and mathematical programing [25].

Overlapping community Some nodes may belong to several communities, so simply assigning the nodes to one community is difficult. Thus, the straightforward solution is to modify the modularity for allowing the nodes belonging to multiple communities at the same time [26–30]. Figure 1 shows two benchmarks about overlapping communities. In Figure 1a, the node v_9 is the overlapping node, and we assign v_9 to community B and C. Thus, we get three communities, and they are $\{\{v_1, v_2, v_3, v_4\}, \{v_5, v_6, v_7, v_8, v_9\}, \{v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}\}\}$. Moreover, v_5 is assigned to A and B in Figure 1b.



(**a**) *G4415*, an example with three communities (**b**) *G415*, an example with two communities

Figure 1. The benchmark with more than two communities and two communities.

In this paper, we focus on the overlapping community detection, and propose the node weight allocation problem denoted by NWA_{OCD} to formulate the community overlap. Since computing the partition with maximum modularity is NP-complete, decreasing the computation cost to seek the near optimal partitions is the popular approach in solving the overlapping community detection. The heuristic algorithms are outstanding in seeking better solutions in large search space, especially for the genetic algorithms (GAs) [2,3,8]. Therefore, some works consider GA as the core approach in their solutions. Mu et al. use a hybrid heuristic approach including GA and the simulated annealing to find out the communities [2]. Shang et al. use GA with an extra local search [3]. The heuristic algorithms perform well in seeking the solution with high quality in a large search space. However, the above results do not deal with the overlapping properties. The overlapping networks have various properties, so some approaches consider the multi-objective approach to find the balanced results [4–6,31]. The balanced results mean that most properties are considered, but the derived results may not be closed to the real-world properties. Therefore, Behera et al. check the similarity between each pair of nodes [8]. The node similarity is also considered by Ezeh et al. to the overlapping nodes and their neighbors [32]. To emphasize the community attribution of each node, Shakya et al. combine fuzzy with the GA to calculate the detail properties of the nodes [7]. Shakya et al. consider the GA to reduce the computation time without decreasing the solution quality too much and adopt the fuzzy communities to identify the overlapping nodes.

Even if some approaches provide the solutions with high modularity, the partitions may not reflect the properties of the real-world networks in some situations. We found that the solution quality could be refined by considering following issues: ignoring overlapping nodes, merging clusters, and reweighting nodes. Therefore, we consider the modularity to design the solution searcher of the approach GA_{IMR}^{NWA}. We firstly modify the fitness function in GA_{IMR}^{NWA} to show the network properties by considering the null model, so the revised fitness function could output the partitions that are closer to the real-world behavior. Moreover, we design three refinement strategies to make the solutions to reflect the real-world properties.

In the simulation, we consider the synthetic network and popular networks that include Zachary Karate Club Network, Books about American Politics, and American College Football to evaluate the solution quality calculated by GA_{IMR}^{NWA} and other approaches. The derived networks correctly reflect the real-world properties in the synthetic networks and the real-world networks. Moreover, the proposed refinement strategies are also evaluated, and the refinement strategies provide higher quality of the derived partitions in the perspective of the real-world behavior. Therefore, the simulation results show that GA_{IMR}^{NWA} outputs the partitions, and the results are closed to the real-world properties.

This paper is organized as follows. The overlapping communities and the problem definition are introduced and formulated in Section 2. The proposed approach GA_{IMR}^{NWA} is shown in Section 3, and the refinement strategies are also listed in this section. The simulation and comparisons are arranged in Section 4, and we show the network partitions in this section. Eventually, the conclusion and future works are stated in Section 5.

2. Preliminary

2.1. Modularity in Overlapping Communities

The community detection of a given network involves two processes. The first one is to find out the network structure and the other one is to determine the numbers of communities. Here we introduce the works proposed by Nepusz et al. [33] to explain the modularity in overlapping communities. Nepusz et al. consider a belonging coefficient matrix $U = [\alpha_{ic}]_{n\times k}$, where n is the number of nodes, and k is a given number of communities. Each entry α_{ic} shows how strongly the node v_i belongs to the community c. The constraint of the relationship between v_i and all communities is:

$$\sum_{c=1}^{k} \alpha_{ic} = 1, \forall \alpha_{ic} \in [0,1], 0 < \sum_{i}^{n} \alpha_{ic} < n. \tag{1}$$

So, the objective function is:

$$D_G(U) = \sum_{i,j=1}^{n} w_{ij}(\tilde{s_{ij}} - s_{ij})^2, \tag{2}$$

where w_{ij} is the predefined weight, $s_{ij} = \sum_{c=1}^{k} \alpha_{ic}\alpha_{jc}$, and $\tilde{s_{ij}}$ is the prior similarity of v_i and v_j. By minimizing Equation (2), the nodes with high similarity will be grouped together. So, U with optimal result $D_G(U)$ is the overlapping community structure.

To determine an appropriate number of communities k, Nepusz et al. iteratively increase the value of k from 2, and then choose the value of k with the highest fuzzy modularity value calculated by Equation (3).

$$Q_{ov}^{Ne} = \frac{1}{2m} \sum_{c=1}^{k} \sum_{i,j=1}^{n} (A_{ij} - \frac{k_i k_j}{2m}) \alpha_{ic}\alpha_{jc} \tag{3}$$

2.2. Problem Definition

The overlapping community detection problem is considered as a node weight allocation problem, denoted by NWA_{OCD} for short. Given a network $G(V, E)$, a maximum number of communities k, and a null model weight γ. Find a modified belonging coefficient matrix $M = [\lambda_{ic}]_{n\times k}$, such that the Q'_{ov} value is maximized. The objective function and constraints are:

$$
\begin{aligned}
\max \quad & Q'_{ov} = \frac{1}{2m}\sum_{c=1}^{k}\sum_{i,j=1}^{n}(A_{ij} - \gamma\frac{k_i k_j}{2m})\lambda_{ic}\lambda_{jc} \\
\text{s.t.} \quad & \lambda_{ic} \in [0, 1] \\
& \sum_{c=1}^{|C|} \lambda_{ic}^{inc_f} = 1.
\end{aligned} \tag{4}
$$

We consider inc_f as the increasing factor. Given $inc_f > 1$, the total weight of an overlapping node over all communities is larger than one, i.e., $\sum_{c=1}^{k} \lambda_{ic} > 1$. The total weight of a non-overlapping node is still equal to one exactly, i.e., $\sum_{c=1}^{k} \lambda_{ic} = 1$.

By solving the NWA_{OCD} problem, the overlapping community structure will be obtained by modifying the optimal solution. Note that if $inc_f = 1$ and $\gamma = 1$, Equation (4) is the same with Equation (5), which means the fuzzy modularity is a special case of the NWA_{OCD} problem.

$$
\begin{aligned}
\max \quad & Q_{ov} = \frac{1}{2m}\sum_{c=1}^{k}\sum_{i,j=1}^{n}(A_{ij} - \frac{k_i k_j}{2m})\alpha_{ic}\alpha_{jc} \\
\text{s.t.} \quad & \alpha_{ic} \in [0, 1] \\
& \sum_{c}^{k} \alpha_{ic} = 1.
\end{aligned} \tag{5}
$$

Although Griechisch et al. [34] apply the fuzzy modularity to find overlapping communities, there are still some networks are unresolved. We introduce the networks with more than two communities and two communities to show this issue. The benchmark is shown in Figure 1. The values of Q_{ov} for G4415 and G415 are shown in Table 1. We can see that v_9 belongs to B in G4415 while v_5 belongs A in G415, and they are not overlapping nodes.

The major difference between Equations (4) and (5) is the coefficient matrix. Each entry in Equation (5) is unweighted while that is weighted in Equation (4). Therefore, we need a mapping as shown in the following equations.

$$
\begin{aligned}
\lambda_{ic} &= \sqrt[inc_f]{\alpha_{ic}} \\
\alpha_{ic} &= \lambda_{ic}^{inc_f}
\end{aligned} \tag{6}
$$

Table 1. The values of Q_{ov} in G4415 and G415.

(**a**) The Q_{ov} values with different assignments of v_9 in G4415.

$\alpha_{9,B}$	$\alpha_{9,C}$	$\alpha_{9,D}$	Q_{ov}
1	0	0	0.5736
0.7	0.3	0	0.5709
0.3	0.7	0	0.5664
0	1	0	0.5624
0	0	1	0.5560

(**b**) The Q_{ov} values with different assignments of v_5 in G415.

$\alpha_{5,A}$	$\alpha_{5,B}$	$\alpha_{5,C}$	Q_{ov}
1	0	0	0.4305
0	0	1	0.4151
0	1	0	0.4058

3. Allocate Node Weight by Genetic Algorithms

Computing the partition with maximum modularity has been proved as the NP-complete problem [13]. Even if we consider the solution with high computation performance, e.g., the cloud computing [35,36] and the parallel computing [37], to compute the partitions for maximizing the modularity, it still requires huge computation resource. Therefore, we propose a GA-based approach to get the near-optimal solution with minimum computation. The proposed algorithm GA_{IMR}^{NWA} includes two steps. We first apply GA to obtain a high-quality feasible solution, and then design three refinement strategies to improve the derived solution to modify the derived partition to be closer to the real-world behavior. In the following context, we will introduce the revised GA algorithm and the refinement strategies.

3.1. Genetic Algorithm

The iterative process of GA as shown in Algorithm 1 includes three major processes: crossover, mutation, and selection. Before invoking the iterative process, the initial population P with $indi_n$ chromosomes will be determined firstly. Each chromosome is represented by $M = [\lambda_{ic}]_{n \times k}$, as shown in Figure 2. Each entry λ_{ic} is a weight to indicate the assignment from v_i to c. The initial population is generated randomly, and each row of M must satisfy the problem constraints. Given a maximum number of iterations max_t, the GA then invokes following processes.

1. **Crossover**: we randomly select two chromosomes C_A and C_B form P, and a random column. The offspring is generated by the selected column of C_B and the remaining part of C_A as shown in Figure 3. The number of offsprings is determined by $indi_n$, and in other words, we will obtain $2 \times indi_n$ chromosomes after the crossover.
2. **Mutation**: the mutation process is launched in 80% probability after finishing the crossover. Once the mutation is invoked, one λ_{ic} of a randomly selected chromosome will be picked up within $[-0.1, 0.1]$. Eventually, the offspring will be normalized to be a feasible solution to fit the requirements in NWA_{OCD}.
3. **Selection**: we consider the modularity to be the objective function, and finding the partition with maximum modularity is the purpose of GA. We use $Q_{ov}^{'}$ to be the fitness function and calculate $Q_{ov}^{'}$ of each solution. Moreover, all chromosomes are sorted in the descending order of $Q_{ov}^{'}$. Computing the chromosomes with maximum $Q_{ov}^{'}$ is the major goal of the GA, so we select top $indi_n$ individuals, and they will survive to the next generation.

Algorithm 1: Genetic algorithm for allocating node weight

Data: max_t: the maximum number of iteration, $indi_n$: the number of survival genes

```
P ← initialization(indi_n);
for t = 1 : max_t do
    P′ ← crossover(P);
    P′ ← mutation(P′);
    P ← selection(P′);
```

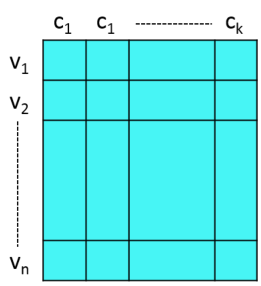


Figure 2. The representation of a chromosome.

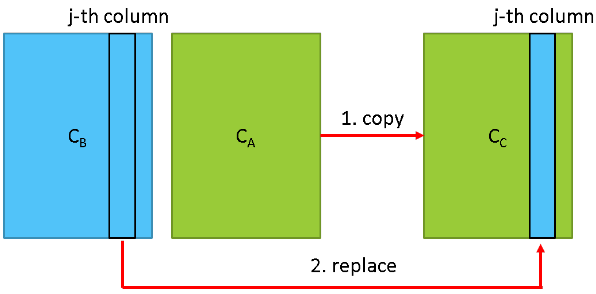


Figure 3. The idea of the crossover operation. Two chromosomes are switched the selected area to generate one offspring.

To keep the heavily overlapping nodes, a threshold α_T in terms of α is given. We transform α_T to the corresponding λ with the threshold λ_T by Equation (6).

3.2. Refinement Strategies

GA provides an elite solution from the population, but this solution may not be suitable for all instances. In the pre-analysis phase, we observed three situations derived by GA_{IMR}^{NWA}, and we could receive better solutions by some extra processes. The situations are (1) lightly overlapping nodes, (2) mergeable clusters, and (3) reweight nodes. We call the processes that are used to get better solutions the "refinement strategies". Therefore, we provide three refinement strategies to refine the solutions for the above situations, respectively.

Ignore slight overlapping nodes The overlapping degree of each λ is important for splitting the communities. Determining the community with low value of λ is easier than that with a higher value. We use a threshold λ_T corresponding to Equation (6) to determine that the entry should be treated as an entry without overlaps. In addition, we also can derive λ_T by Equation (6). When $\lambda < \lambda_T$, we set λ as zero. When λ_T is set as a higher value, more entries will be assigned to single community.

Merge clusters Some small communities should be merged by other large community. If the overlapping ratio of any two communities is larger than a given merge threshold m_T, they should be simply merged to a single community. Given two non-empty communities, we define $ov_{ratio} = |C_1 \cap C_2| / \min(|C_1|, |C_2|)$ to be the overlapping ratio. When ov_{ratio} is larger than a given threshold, C_1 and C_2 will be merged.

Reweight node values To calculate the weight distribution of each overlapping node, directly converting λ to α via Equation (6) results in a situation that a node belongs to multiple communities but the majority of its weight is allocated to one community. To avoid this problem, we propose the reweight strategy. The weight should be proportional to the number of edges that v_i linked in c. Moreover, if the neighbors of v_i in c are more than the average number of nodes in c, c is more important than others for v_i. Given a community c, $avgNighbor_c = \sum_{i,j \in V(c)} A_{ij} / |V(c)|$ represents the average number of neighbors and $\alpha_i = \sum_{c \in C(i)} \sum_{j \in V(c)} A_{ij} / avgNighbor_c$ be the normalized term. Therefore, we have the new weight is:

$$\alpha_{ic} = \frac{1}{\alpha_i} \sum_{j \in V(c)} \frac{A_{ij}}{avgNighbor_c}, \tag{7}$$

where $V(c)$ is the set of nodes belong to c and $C(i)$ is the set of communities that v_i belongs to. We use α_i for normalization, so we have $\sum_{c=1}^{k} \alpha_{ic} = 1$.

4. Simulations

We consider a synthetic network and three real networks including Zachary Karate Club network, Books about American Politics, and American College Football to evaluate the performance of GA_{IMR}^{NWA}.

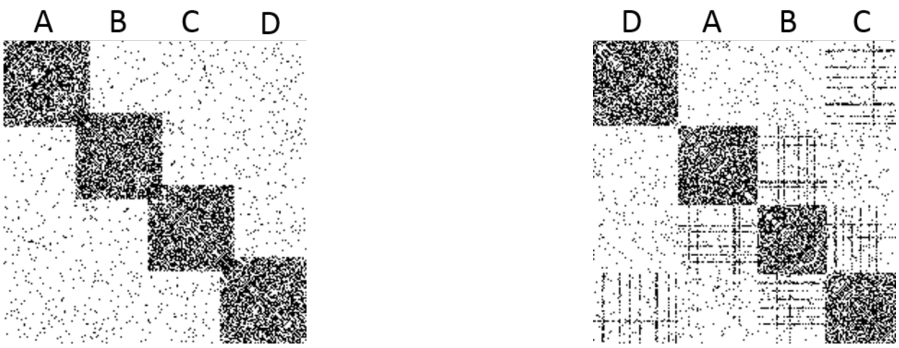
The evaluation criteria involve detecting overlapping community structure, detecting meaningful communities, detecting dense overlaps, and detecting heavily overlapping nodes.

4.1. Synthetic Network

We consider *G210* as our synthetic network which has 210 nodes and four pre-defined communities A, B, C and D. Each of them has 60 nodes and 10 shared by any two continuous communities, i.e., $A = \{v_1 : v_{60}\}$, $B = \{v_{51} : v_{110}\}$, $C = \{v_{101} : v_{160}\}$, and $D = \{v_{151} : v_{210}\}$. Note that A and B share nodes $\{v_{51}, \ldots, v_{60}\}$, B and C share nodes $\{v_{101}, \ldots, v_{110}\}$ and so on. Each pair of nodes has 3% chances to be linked to each other, and for each community they shared, an extra 55% chances for them to be linked. Thus, overlapping parts will be denser than non-overlapping parts [38].

Since the fuzzy modularity is a special case of the NWA_{OCD} problem, we could use the same optimization strategy to solve the problem. The parameter settings are $inc_f = 1.5$ and 1, $\alpha_T = 0$, $m_T = -1$, $k = 6$, and $\gamma = 1$. Figure 4 shows the bitmaps of sorted adjacency matrices. The black and white points represent the entries of 1s and 0s respectively. The adjacency matrices are sorted by the following strategy:

1. Nodes are grouped by the detected community id. For the overlapping nodes, only the smallest id is counted.
2. For each c, all nodes are sorted in descending order of λ_{ic}. Therefore, the overlapping nodes will be in the bottom area of each community.



(**a**) Result of our method (**b**) Result of fuzzy modularity

Figure 4. The comparison between GA_{IMR}^{NWA} and fuzzy modularity.

Figure 4a is the result obtained by GA_{IMR}^{NWA}. The dense blocks indicate four communities, and two continuous blocks have an overlapping part which is composed of overlapping nodes. In this result, all the overlapping and non-overlapping nodes are correctly identified. Figure 4b is the result of fuzzy modularity. Four communities are detected too, but no overlapping nodes are identified.

Although the maximum number of communities is six, only four communities were detected while the other two were empty communities. Since the number of communities could be captured by modularity [39], it is unnecessary to know the exactly value of number of communities in our method.

4.2. Zachary Karate Club Network

Zachary karate club network [40] is a popular benchmark for community detection algorithms. It has 34 nodes and 78 edges while nodes are members and edges are friendships between them. This network includes two groups due to a disagreement between the administrator and the instructor. Figure 5 is the result captured by the fuzzy modularity. In this experiment, we evaluate the results with different inc_f settings, and show the importance of "ignore slight overlapping nodes" and "reweight node values". Finally, we apply our method on the case with the value $k = 2$, and halved the null model.

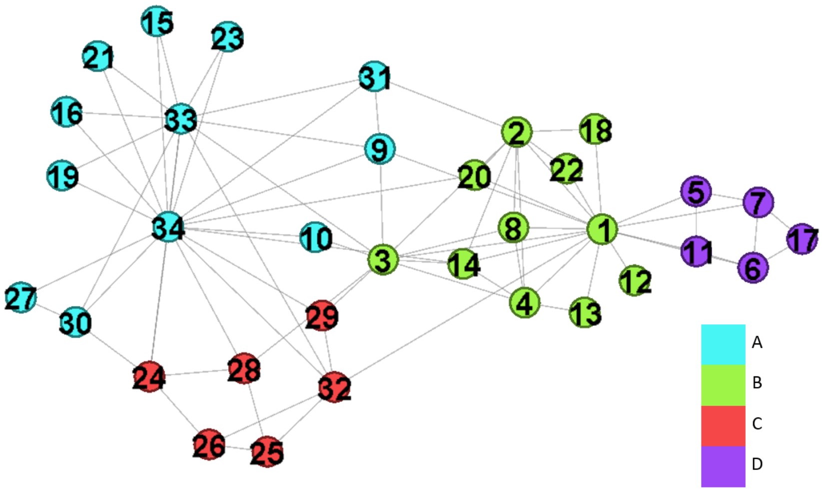


Figure 5. Detected communities of the karate network by fuzzy modularity.

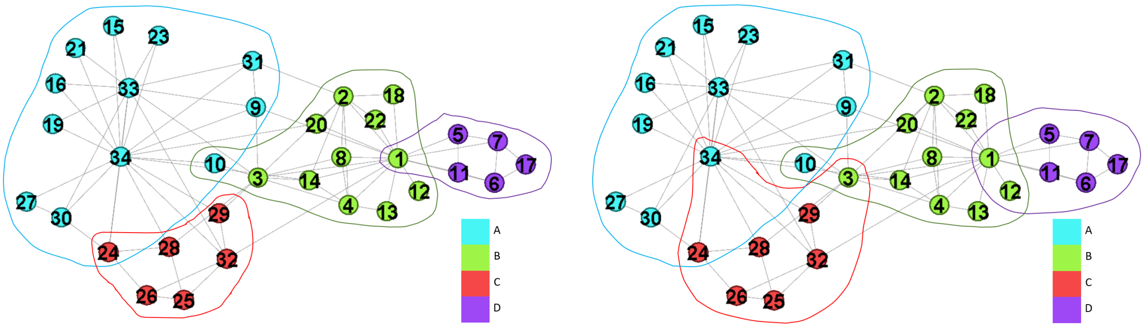
4.2.1. Effects of Weight Increasing Factor

We first evaluate the communities captured by GA_{IMR}^{NWA} in the networks with $inc_f = \{1, 1.2, 1.5, 1.7\}$ while $\alpha_T = 0.01$, $m_T = -1$, $k = 8$, and $\gamma = 1$. The corresponding $Q_{ov}^{'} = \{0.419, 0.422, 0.427, 0.430\}$. We consider the fuzzy modularity with $inc_f = 1$ as our baseline since it outputs the correct solution.

Figure 6a is the result with $inc_f = 1.2$, and we get four communities and three overlapping nodes while λ is shown in Table 2a. The network separation in Figure 6a is identical to that in Figure 5, but maximizing the modularity outputs a larger one than that we derived. When inc_f is increased from 1.2 to 1.5, we get two extra overlapping nodes, and they are v_{12} and v_{34}. When inc_f is set as 1.7, the values of λ are changes as shown in Table 2c, and others are identical to that derived by $inc_f = 1.5$. Therefore, larger settings of inc_f results in more overlapping nodes.

Table 2. The comparison with various inc_f settings.

(**a**) λ values of overlapping nodes in Figure 6a with $inc_f = 1.2$				
Node	λ_{iA}	λ_{iB}	λ_{iC}	λ_{iD}
v_1		0.967		0.068
v_{10}	0.747	0.362		
v_{24}	0.419		0.696	
(**b**) λ values of overlapping nodes in Figure 6b with $inc_f = 1.5$				
Node	λ_{iA}	λ_{iB}	λ_{iC}	λ_{iD}
v_1		0.917		0.246
v_3		0.986	0.075	
v_{10}	0.700	0.556		
v_{12}		0.993		0.048
v_{24}	0.553		0.703	
v_{34}	0.993		0.048	
(**c**) λ values of overlapping nodes in Figure 6b with $inc_f = 1.7$				
Node	λ_{iA}	λ_{iB}	λ_{iC}	λ_{iD}
v_1		0.888		0.369
v_3		0.987	0.108	
v_{10}	0.694	0.636		
v_{12}		0.926		0.290
v_{24}	0.600		0.726	
v_{34}	0.989		0.097	



(**a**) Detected communities with $inc_f = 1.2$ (**b**) Detected communities with $inc_f = 1.5$ and 1.7

Figure 6. The communities detected by GA^{NWA}_{IMR} under various inc_f settings.

Considering that a node has only one edge connecting to an overlapping node, e.g., v_{12}, the isolation has the same property with that held by the overlapping node. Moreover, we found that $Q^{'}_{ov}$ derived by GA^{NWA}_{IMR} is higher than the optimal Q. It implies that the overlapping structure is easier to be detected as assigning higher weight to the overlapping nodes.

Here we consider an extreme case that all nodes are overlapped, i.e., $inc_f = 4$. We analyze the obtained result, and then find the "duplicate communities". Two or more communities are extremely overlapped with each other, and even some of them are just the same community.

Figure 7 shows the fuzzy partition result. Four communities are detected, but two of them denoted by dotted lines are the subsets of the rest two communities denoted by solid lines. Therefore, two sets should be merged to a correct community. After merging the communities, we derive two communities, and there is only one overlapping node v_{10}. However, the value of $Q^{'}_{ov}$ is decreased from 0.526 to 0.371 simultaneously.

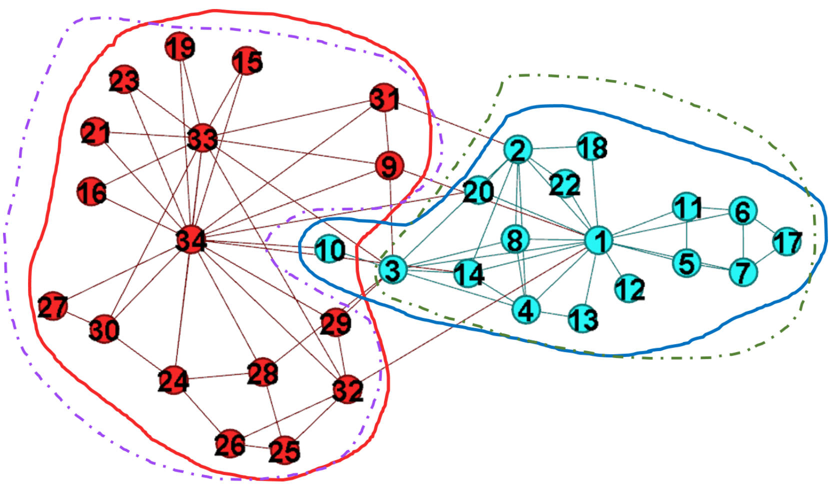


Figure 7. Duplicate communities result.

Even if we derive the result with maximized value of $Q^{'}_{ov}$, the solution does not show the correct properties of the communities. We use the refinement strategies to get the solution with lower quality but more closed to the real-world properties. Therefore, the refinement strategies are useful for improving the solution quality in terms of the real-world consideration.

4.2.2. Effects of Ignoring Slight Overlapping Nodes

We consider the network with $inc_f = 1.5$ to evaluate the effects of the *ignore* step. The result with and without the *ignore* step are 0.427114 and 0.427117, respectively. Figure 8 and Table 3 are the detected communities and values of λ. Two overlapping nodes v_{28} and v_{30} are ignored. Since most of their weights were kept in a specific community, reducing the weights will not decrease $Q^{'}_{ov}$

dramatically. Therefore, the process of ignoring slight overlapping nodes helps to keep those heavily overlapping nodes.

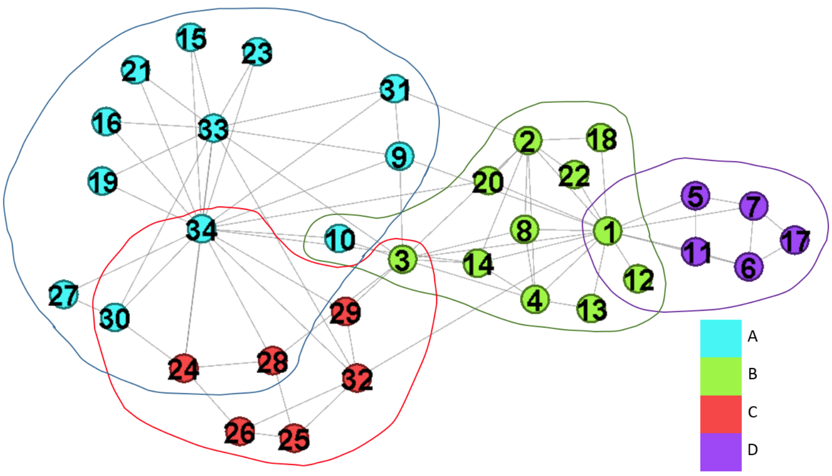


Figure 8. Detected communities with $inc_f = 1.5$ (before ignoring).

Table 3. λ values of overlapping nodes in Figure 8 with $inc_f = 1.5$ (before ignoring).

Node	λ_{iA}	λ_{iB}	λ_{iC}	λ_{iD}
v_1		0.917		0.246
v_3		0.986	0.075	
v_{10}	0.700	0.556		
v_{12}		0.993		0.048
v_{24}	0.553		0.703	
v_{28}	0.002		0.999	
v_{30}	0.999		0.004	
v_{34}	0.993		0.048	

4.2.3. Effects of Reweight Strategy

To emphasize the importance of the communities, we propose a reweight strategy to assign various weights. The result with reweight strategy is identical to that shown in Figure 6b. Table 4a,b show the value of λ without and with considering the reweight strategy, respectively. The reweight strategy reduces the gap of the number of edges for connecting the inside-community nodes and outside-community nodes. However, the structure of the main community may be changed after reweighting, because the values are inversely proportional to the average number of neighbors in the communities to that out of communities. For example, v_{12} is unbalanced before reweighting, but the value of λ of v_{12} reflect the real-world behavior.

4.2.4. The Network with Two-Communities

We examine the network with exactly two communities to verify the property illustrated in Figure 1b can be captured by GA_{IMR}^{NWA}. We consider $inc_f = 1.5$, $\alpha_T = 0.01$, $m_T = -1$, $k = 2$, and $\gamma = 0.5$. In this case, we easily find out the overlapping nodes. The results are shown in Figure 9 and Table 5.

GA_{IMR}^{NWA} derives three overlapping nodes as shown in Table 5. From Figure 9, we have $Q_{ov}^{'} = 0.628$, and the dotted curve is the real split of the club network. v_3 is the main overlapping node since it has a roughly balanced weight value. In summary, the two-community problem is solved by reducing the number of expected edges.

Table 4. The comparison of GA_{IMR}^{NWA} with reweighting and without reweighting.

(a) Before reweighting				
Node	λ_{iA}	λ_{iB}	λ_{iC}	λ_{iD}
v_1		0.917		0.246
v_3		0.986	0.075	
v_{10}	0.700	0.556		
v_{12}		0.993		0.048
v_{24}	0.553		0.703	
v_{34}	0.993		0.048	
(b) After reweighting				
Node	α_{iA}	α_{iB}	α_{iC}	α_{iD}
v_1		0.611		0.389
v_3		0.709	0.291	
v_{10}	0.52	0.48		
v_{12}		0.440		0.560
v_{24}	0.468		0.532	
v_{34}	0.725		0.275	

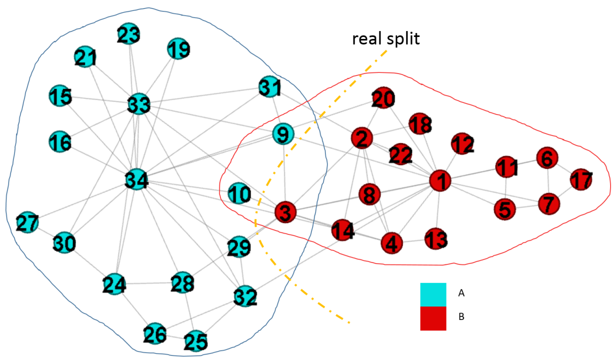


Figure 9. Detected communities with $k = 2$, and $\gamma = 0.5$

Table 5. λ values of overlapping nodes in Figure 9.

Node	λ_{iA}	λ_{iB}
v_3	0.493	0.753
v_9	0.987	0.071
v_{10}	0.984	0.085

4.2.5. Compare with Different Algorithms

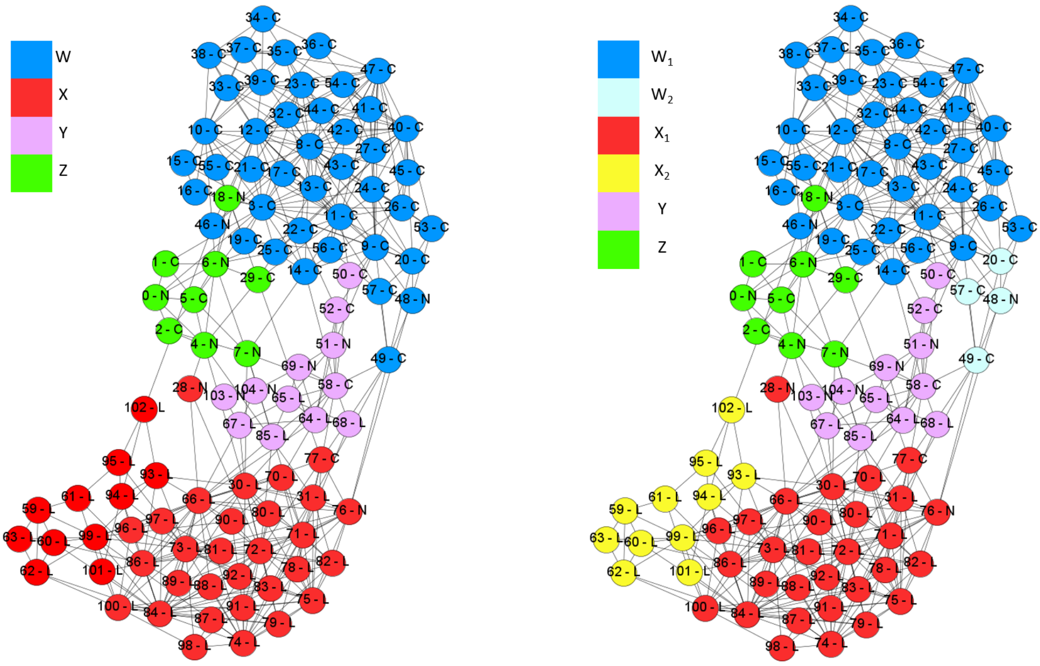
In the above simulations, GA_{IMR}^{NWA} detects two communities, and we compare the result with previous algorithms in this dataset. Shen et al. captured three overlapping communities [30], and the overlapping nodes are v_1, v_3 and v_9. However, v_{12} is missed in the method of Shen et al. The property of the overlapping communities in v_{12} is not discovered. The node v_{12} has exactly one neighbor that is node v_1, so v_{12} should have the same overlapping properties as that of v_1.

Chen et al. captured two overlapping communities [29], and their results are similar to ours as shown in Figure 9. Chen et al. found one overlapping node v_{10}. Node v_{10} has two edges that one connects to the left community while the other one comments to the right community. Therefore, considering v_{10} as the overlapping node is reasonable. However, the node v_3 has five edges where three edges connect to the left community while two connect to the right community. v_3 is more appropriate than v_{10} to be the overlapping node.

From the above observation, the communities are split more precisely by GA_{IMR}^{NWA} than the previous works. For the considerations of the split appropriateness, e.g., the number of detected communities, and the split correctness, e.g., the overlapping nodes, GA_{IMR}^{NWA} provides more precise results than other approaches.

4.3. Books about American Politics

This network is built from the transaction data from amazon.com [41]. The network has 105 nodes and 441 edges while nodes indicate books and edges are frequent co-purchase events. The nodes are labeled by three categories including *liberal*, *neutral*, or *conservative*. Each category has 43, 13, and 49 nodes respectively. In this simulation, we consider $inc_f = 1.5$, $\alpha_T = 0.01$, $m_T = 0.5$, $k = 8$, and $\gamma = 1$. We evaluate the performance of the merge strategy. Figure 10a,b are the solutions with and without merge strategies respectively. The text on each node is the node id and the real label. The results of Q'_{ov} are 0.528 and 0.533 for the results with and without merge strategy.



(**a**) Book communities after merging (**b**) Book communities before merging

Figure 10. The book comparison between GA_{IMR}^{NWA} with merging and without merging.

4.3.1. The Result with Merge Strategy

GA_{IMR}^{NWA} with the merge strategy detects four communities denoted by W, X, Y, and Z. Most nodes belong to two large communities W and X, which are mainly consisted of *conservative* and *liberal* books respectively. Most *neutral* books belong to two small communities. This result is similar to that obtained by Newman [39]. Table 6 is the values of λ for ten overlapping nodes. There are four *neutral* nodes, that is 40% of all overlapping nodes and 30% of all *neutral* nodes. The result implies that *neutral* books are often co-purchased with different books.

4.3.2. The Result without Merge Strategy

GA_{IMR}^{NWA} without the merge strategy splits W and X into two parts respectively denoted by W_1, W_2, X_1 and X_2. A small community including v_{48}, v_{49} and v_{57} has been detected by the modularity maximization [25]. Therefore, we also found this community and labeled it by W_2.

Moreover, we also detect an extra community X_2. After analyzing the edge density of X_1 and X_2, they are both denser than the merged community X. Besides, the overlapped part is even denser as shown in Table 7. The density function definition is as follows:

$$D(c) = \frac{1}{\binom{|V(c)|}{2}} \sum_{i,j \in V(c)} \frac{A_{ij}}{2}. \quad (8)$$

Table 6. λ values of overlapping nodes in Figure 10a.

Node	λ_{iW}	λ_{iX}	λ_{iY}	λ_{iZ}	Label
v_3	0.986			0.076	conservative
v_7			0.254	0.913	neutral
v_9	0.975		0.11		conservative
v_{18}	0.528			0.724	neutral
v_{22}	0.955			0.164	conservative
v_{25}	0.922			0.236	conservative
v_{28}		0.72		0.533	neutral
v_{46}	0.921	0.238			neutral
v_{50}	0.458		0.781		conservative
v_{85}			0.981	0.093	liberal

Table 7. Density value of each part of community X.

	X	X_1	X_2	$X_1 \cap X_2$
$D(c)$	0.20	0.27	0.34	0.63

The overlapping ratios of (W_1, W_2) and (X_1, X_2) are 57% and 53%, respectively. High overlapping ratios indicate that we could merge each pair of them without decreasing Q'_{ov} too much. Therefore, modularity can not detect X_2 because of high overlapping ratio and dense overlapped part. This result shows the dense overlaps can be discovered by GA_{IMR}^{NWA} correctly.

4.4. American College Football

This is the network of American football games between Division IA colleges in 2000 [42]. It has 115 nodes, 613 edges and 12 conferences as shown in Table 8. Nodes are teams and edges are games between the corresponding two teams while nodes are labeled by the conferences they belong to. We apply $inc_f = 1.5$, $\alpha_T = 0.01$, $m_T = -1$, $k = 15$, and $\gamma = 1$ in this simulation.

Table 8. Labels of conferences.

Label	Conference	#Teams	Label	Conference	#Teams
0	Atlantic Coast	9	**6**	Mid-American	13
1	Big East	8	**7**	Mountain West	8
2	Big Ten	11	**8**	Pacific Ten	10
3	Big Twelve	12	**9**	Southeastern	12
4	Conference USA	10	**10**	Sun Belt	7
5	Independents	5	**11**	Western Athletic	10

Figure 11 shows the result with $Q'_{ov} = 0.607$, true labels are on the nodes. Ten communities and 17 overlapping nodes are detected. Most conferences are well matched to the detected communities except for the conferences *Independents* (Label 5) and *Sun Belt* (Label 10). There are total seven overlapping nodes in these two conferences. From Table 9, 41% overlapping nodes and 58% nodes are in the two conferences.

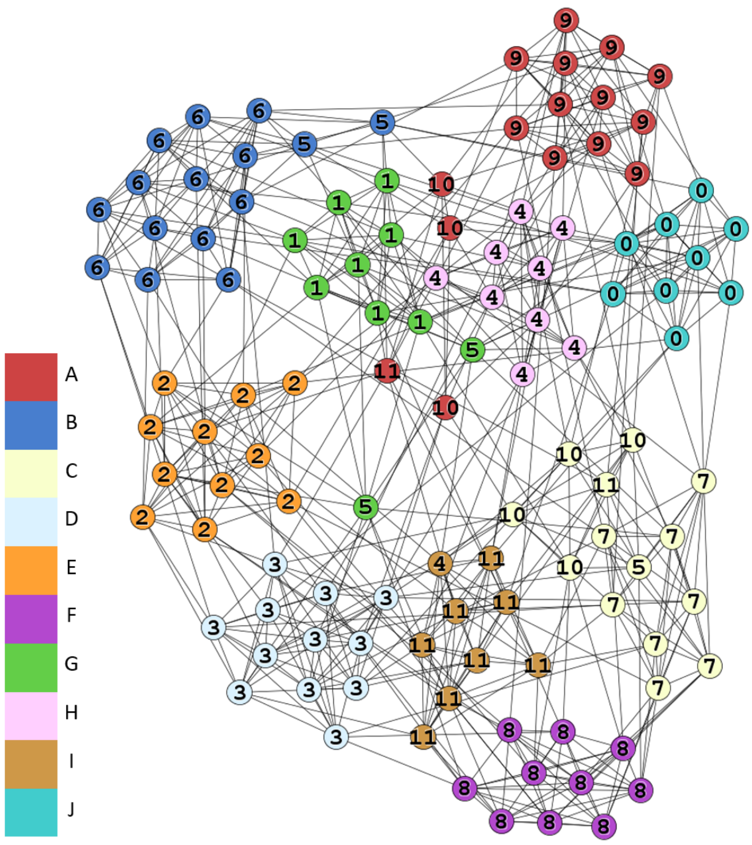


Figure 11. Football communities.

Table 9. λ values of overlapping nodes in Figure 11.

Node	λ_{iA}	λ_{iB}	λ_{iC}	λ_{iD}	λ_{iE}	λ_{iF}	λ_{iG}	λ_{iH}	λ_{iI}	λ_{iJ}	Label
v_2				0.06	0.99						**2**
v_8			0.058			0.991					**8**
v_9			0.971			0.123					**7**
v_{11}			0.937	0.204							**10**
v_{23}			0.981			0.094					**7**
v_{36}	0.575	0.682									**5**
v_{44}							0.11	0.975			**4**
v_{50}			0.902			0.274					**10**
v_{58}	0.961								0.149		**11**
v_{66}	0.067							0.989			**4**
v_{67}			0.05						0.993		**11**
v_{69}			0.992						0.054		**10**
v_{78}			0.05			0.992					**8**
v_{80}							0.941	0.121		0.126	**5**
v_{82}				0.065	0.082	0.097	0.953				**5**
v_{97}	0.704			0.326				0.368			**10**
v_{112}	0.065							0.989			**4**

The conference *Independents* has five teams, and only one game was held. This is the major reason that makes this conference undetectable. However, the teams often play with other teams in varied conferences, and this phenomenon results in the overlapping property. For example, v_{82} is assigned to four communities, although it connected to community G with four edges. v_{82} still connects to other three communities with a significant number of edges, so that is why it belongs to many communities simultaneously as shown in Figure 12. On the other hand, *Sun Belt* is in the similar situation. In this example, the heavily overlapping nodes could be detected by our method.

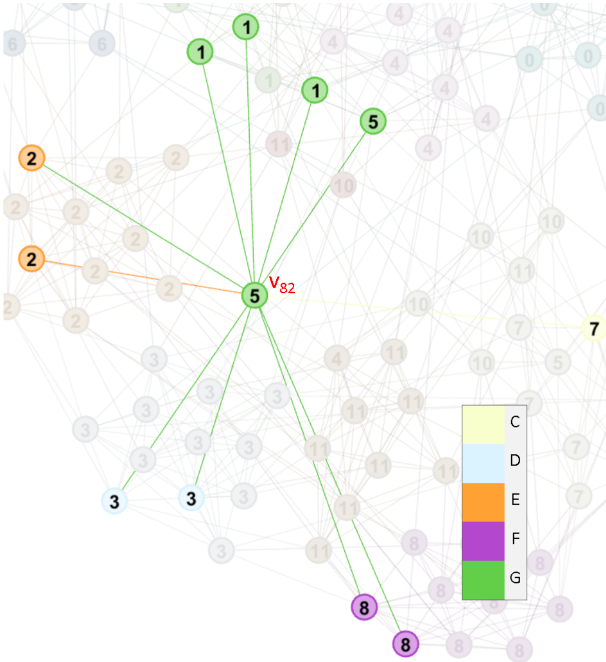


Figure 12. The node v_{82} and its neighbors in football network.

4.5. Dolphin Network

The Dolphin Network is a common benchmark for evaluating the overlapping communities. Some results consider the Dolphin Network to evaluate the community quality [26,43]. We compare the proposed GA_{IMR}^{NWA} with related results in this simulation. The Dolphin Network includes 62 nodes and 159 edges, and two communities are detected eventually for a long-term observation.

The distribution of λ for overlapping nodes is listed in Table 10 while the separation with $Q' = 0.535$ is illustrated in Figure 13. According to the refinement strategy Ignore slight overlapping nodes, we get three overlapping nodes v_{20}, v_{28}, and v_{44} after decreasing the setting of λ_T from 1.0 to 0.9. The overlapping nodes are marked by the red circle with dot lines, and they are marked by the overlapping nodes based on the distribution of λ. On the other hand, we also consider $m_T = -1$ in Dolphin network as the same setting in the above simulations. The community B, C, D, and E are merged according to the refinement strategy Merge clusters. Eventually, we get two communities.

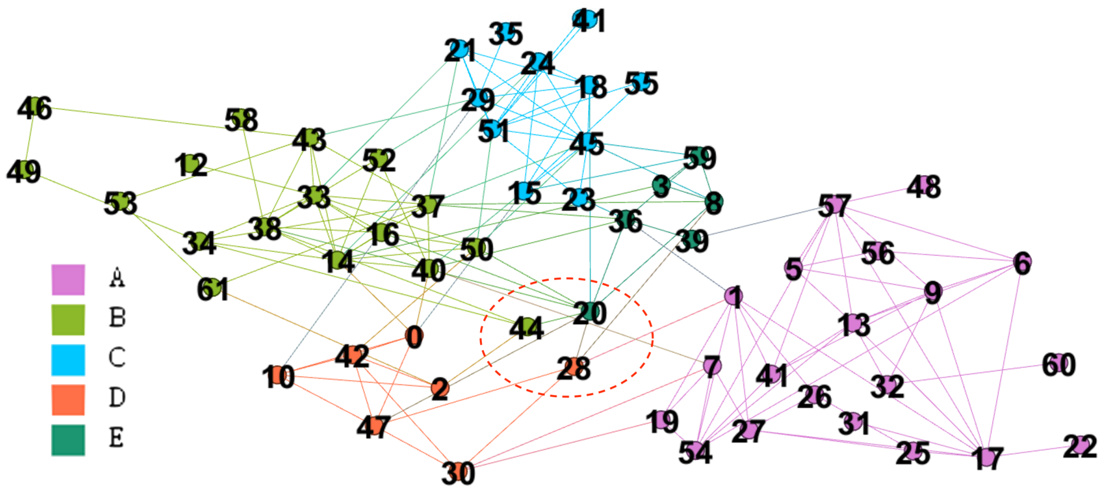


Figure 13. Five communities are detected by the proposed approach. There are three overlapping nodes when using $\lambda_T = 0.9$. Therefore, the community B, C, D, and E could be merged by refinement strategy Ignore slight overlapping nodes, and we find two communities eventually.

Nicosia et al. found four communities in Dolphin network [26]. The overlapping nodes are mentioned, but the authors did not list the overlapping nodes. Wang and Fleury provided detail analysis and found two communities from Dolphin network with $Q = 0.385$ [43]. The separation is acceptable, but the network structure is not so strong comparing to Figure 13. After considering the refinement strategies, the separation derived by the proposed GA_{IMR}^{NWA} is similar to that provided by Wang and Fleury in [43], but the structure of our network is stronger than the network in [43]. In summary, the refinement strategies are useful in revising the network separation to be closer to the real-world behavior, and the strength of the network structure is also improved.

Table 10. λ values of overlapping nodes in Figure 13.

Node	λ_{iA}	λ_{iB}	λ_{iC}	λ_{iD}	λ_{iE}
v_0		0.008		0.999	
v_1	0.998				0.023
v_2		0.076		0.986	
v_7	0.990			0.061	
v_8				0.022	0.998
v_{15}			0.999	0.000	0.013
v_{19}	0.986			0.074	
v_{20}		0.361		0.364	0.682
v_{23}			0.909		0.261
v_{28}				0.823	0.400
v_{30}	0.051			0.992	
v_{36}		0.013			0.999
v_{37}		0.990			0.062
v_{39}	0.255				0.912
v_{40}		0.998			0.021
v_{44}		0.844		0.362	0.038
v_{45}			0.992		0.053
v_{47}				0.999	0.011
v_{50}		0.928	0.175	0.103	
v_{52}		0.999	0.009		
v_{59}			0.138		0.965
v_{61}		0.925		0.229	

5. Conclusion and Discussion

Given a network, the modularity is used for measuring the partition quality while the fuzzy clustering recognizes the overlapping communities. Combining above concepts together to be the fuzzy modularity is an appropriate method to formulate the structure of the given network with overlapping communities. Maximizing the modularity outputs the partition with well network structure, but computing the partition with maximum modularity requires huge computation cost. Therefore, the heuristic algorithms are outstanding in seeking high quality solution from a large search space, and we can find some research results of using heuristic algorithms for finding the partitions with maximum modularity. However, there are some special cases that we have to deal with. We find out three common situations from the partitions derived from the GA with modularity maximization and propose three solution refinement strategies to ignore overlapping nodes, merge clusters, and reweight nodes to separate the network to be closer the real-world behaviors. Moreover, we modify the fitness function of the GA to consider the null model for measuring the distance between the derived partition and the random graph. Thus, the simulation results show that the proposed GA_{IMR}^{NWA} provide significant improvement comparing with previous approaches. The derived partition may not always have maximum modularity, but the community structure is more reasonable than the partitions derived by previous works. GA_{IMR}^{NWA} measures the connectivity of nodes and reweight the overlapping nodes to reflect the correct properties in the given networks. Eventually, GA_{IMR}^{NWA}

determines the partitions appropriately, but the heavily overlapping nodes may be marked as the interior nodes by other approaches.

The overlapping nodes could be detected and provided appropriate allocation by GA_{IMR}^{NWA}. During the simulations, we found some extension works that will be address in the future, and they are listed as follows:

1. In our simulations, we got an interesting result as shown in Figure 14 from the karate network with $inc_f = 2$. The result consists of three communities, and they are grouped by v_{33}, v_3 and v_1. The community with v_3 that the nodes are marked by red could be consider as an overlapping set. It means that the networks not only have overlapping nodes but also overlapping groups. Thus, applying the fuzzy concept to the communities will eliminate the group with v_3, and they may be more closed to the real-world behavior. Since the members in the group with v_3 may belong to different communities based on the situations, e.g., the competitions or the events. Therefore, assigning the red nodes to any community may be inappropriate.
2. The proposed algorithm invokes GA to compute the preliminary partitions and then adopts proposed refinement strategies to correct the partitions by the secondary processes. The refinement strategies could be considered as the local search to improve the partition quality in each iteration. However, it is a tradeoff between the computation cost and the partition quality. Once the refinement strategies are modified from the external processes to the internal processes in GA, the computation cost will be increased. Moreover, the given networks may not always consist of the target properties that could be improved by the refinement strategies. Therefore, the refinement strategies could be designed as local search approaches, but the trigger of launching the local search approaches should be analyzed in the future.

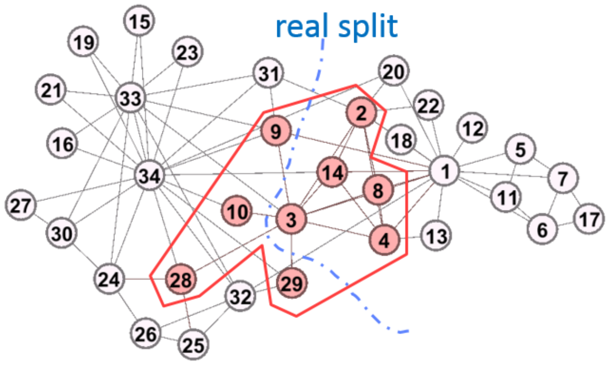


Figure 14. The 5th detected community of the karate network.

Author Contributions: Conceptualization, C.-K.T.; Data curation, C.-Y.C. and T.-W.C.; Project administration, S.-L.L.; Software, C.-Y.C. and T.-W.C.; Supervision, S.-L.L.; Writing—original draft, C.-K.T. and H.-J.H.; Writing–review & editing, C.-K.T. and H.-J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Technology of the Republic of China grant number 108-2221-E-194-024- and MOST 108-2221-E-167-022-.

Acknowledgments: The authors would like to thank all reviewers for giving the constructive comments towards improving this paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

GA Genetic Algorithms

References

1. Rosso, M.A.; McClelland, M.K.; Jansen, B.J.; Fleming, S.W. Using Google AdWords in the MBA MIS course. *J. Inf. Syst. Educ.* **2019**, *20*, 6.
2. Mu, C.H.; Xie, J.; Liu, Y.; Chen, F.; Liu, Y.; Jiao, L.C. Memetic algorithm with simulated annealing strategy and tightness greedy optimization for community detection in networks. *Appl. Soft Comput.* **2015**, *34*, 485–501. [CrossRef]
3. Shang, R.; Bai, J.; Jiao, L.; Jin, C. Community detection based on modularity and an improved genetic algorithm. *Physica A* **2013**, *392*, 1215–1231. [CrossRef]
4. Bello-Orgaz, G.; Salcedo-Sanz, S.; Camacho, D. A multi-objective genetic algorithm for overlapping community detection based on edge encoding. *Inf. Sci.* **2018**, *462*, 290–314. [CrossRef]
5. Li, Z.; Liu, J. A multi-agent genetic algorithm for community detection in complex networks. *Physica A* **2016**, *449*, 336–347. [CrossRef]
6. Yuxin, Z.; Shenghong, L.; Feng, J. Overlapping community detection in complex networks using multi-objective evolutionary algorithm. *Comput. Appl. Math.* **2017**, *36*, 749–768. [CrossRef]
7. Shakya, H.K.; Singh, K.; Biswas, B. An efficient genetic algorithm for fuzzy community detection in social network. In Proceedings of the International Conference on Advanced Informatics for Computing Research, Punjab, India, 17–18 March 2017.
8. Behera, R.K.; Naik, D.; Rath, S.K.; Dharavath, R. Genetic algorithm-based community detection in large-scale social networks. *Neural Comput. Appl.* **2020**, *32*, 9649–9665. [CrossRef]
9. Binesh, N.; Rezghi, M. Fuzzy clustering in community detection based on nonnegative matrix factorization with two novel evaluation criteria. *Appl. Soft Comput.* **2018**, *69*, 689–703. [CrossRef]
10. Naderipour, M.; Zarandi, M.H.F.; Bastani, S. A type-2 fuzzy community detection model in large-scale social networks considering two-layer graphs. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103206. [CrossRef]
11. Yang, C.T.; Chen, S.T.; Den, W.; Wang, Y.T.; Kristiani, E. Implementation of an intelligent indoor environmental monitoring and management system in cloud. *Future Generat. Comput. Syst.* **2019**, *96*, 731–749. [CrossRef]
12. Newman, M.E.J.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. [CrossRef] [PubMed]
13. Brandes, U.; Delling, D.; Gaertler, M.; Goerke, R.; Hoefer, M.; Nikoloski, Z.; Wagner, D. Maximizing Modularity is hard. *arXiv* **2006**, arXiv:physics/0608255.
14. Lai, D.; Lu, H.; Nardini, C. Enhanced modularity-based community detection by random walk network preprocessing. *Phys. Rev. E* **2010**, *81*, 066118. [CrossRef] [PubMed]
15. Huang, J.; Sun, H.; Han, J.; Deng, H.; Sun, Y.; Liu, Y. SHRINK: A Structural Clustering Algorithm for Detecting Hierarchical Communities in Networks. In Proceedings of the 19th Conference on Information and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010.
16. Dinh, T.N.; Thai, M.T. Community detection in scale-free networks: approximation algorithms for maximizing modularity. *IEEE J. Select. Areas Commun.* **2013**, *31*, 997–1006. [CrossRef]
17. Fortunato, S.; Barthélemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 36–41. [CrossRef]
18. Arenas, A.; Fernández, A.; Gómez, S. Analysis of the structure of complex networks at different resolution levels. *New J. Phys.* **2008**, *10*, 053039. [CrossRef]
19. Newman, M.E.J. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *69*, 066133. [CrossRef]
20. Clauset, A.; Newman, M.E.J. Moore, C. Finding community structure in very large networks. *Phys. Rev. E* **2004**, *70*, 066111. [CrossRef]
21. Newman, M.E.J. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **2006**, *74*, 036104. [CrossRef]
22. White, S.; Smyth, P. A spectral clustering approach to finding communities in graph. In Proceedings of the SIAM International Conference on Data Mining, Beach, CA, USA, 21–23 April 2005.
23. Richardson, T.; Mucha, P.J.; Porter, M.A. Spectral Tripartitioning of Networks. *Phys. Rev. E* **2009**, *80*, 0036111. [CrossRef]

24. Guimera, R.; Amaral, L.A.N. Functional cartography of complex metabolic networks. *Nature* **2005**, *433*, 895–900. [CrossRef]
25. Agarwal, G.; Kempe, D. Modularity-maximizing graph communities via mathematical programming. *EPJB* **2008**, *66*, 409–418. [CrossRef]
26. Nicosia, V.; Mangioni, G.; Carchiolo, V.; Malgeri, M. Extending the definition of modularity to directed graphs with overlapping communities. *J. Stat. Mech* **2009**, *2009*, 03024. [CrossRef]
27. Reichardt, J.; Bornholdt, S. Statistical mechanics of community detection. *Phys. Rev. E* **2006**, *74*, 016110. [CrossRef] [PubMed]
28. Liu, J. Fuzzy modularity and fuzzy community structure in networks. *Eur. Phys. J. B* **2010**, *77*, 547–557. [CrossRef]
29. Chen, D.; Shang, M.; Lv, Z.; Fu, Y. Detecting overlapping communities of weighted networks via a local algorithm. *Physica A* **2010**, *389*, 4177–4187. [CrossRef]
30. Shen, H.-W.; Cheng, X.-Q.; Guo, J.-F. Quantifying and identifying the overlapping community structure in networks. *J. Stat. Mech.* **2009**, *2009*, 07042. [CrossRef]
31. Choong, J.J.; Liu, X.; Murata, T. Optimizing Variational Graph Autoencoder for Community Detection with Dual Optimization. *Entropy* **2020**, *22*, 197. [CrossRef]
32. Ezeh, C.; Tao, R.; Zhe, L.; Yiqun, W.; Ying, Q. Multi-Type Node Detection in Network Communities. *Entropy* **2019**, *21*, 1237. [CrossRef]
33. Nepusz, T.; Petróczi, A.; Nógyessy, L.; Bazsó, F. Fuzzy communities and the concept of bridgeness in complex networks. *Phys. Rev. E* **2008**, *77*, 016107. [CrossRef]
34. Griechisch, E.; Pluhár, A. Community detection by using the extended modularity. *Acta Cybern.* **2011**, *20*, 69–85. [CrossRef]
35. Yang, C.T.; Shih, W.C.; Chen, L.T.; Kuo, C.T.; Jiang, F.C.; Leu, F.Y. Accessing medical image file with co-allocation HDFS in cloud. *Future Generat. Comput. Syst.* **2015**, *43*, 61–73. [CrossRef]
36. Yang, C.T.; Liu, J.C.; Chen, S.T.; Lu, H.W. Implementation of a big data accessing and processing platform for medical records in cloud. *J. Med. Syst.* **2017**, *41*, 149. [CrossRef] [PubMed]
37. Natarajan, S.; Vairavasundaram, S.; Ravi, L. Optimized fuzzy-based group recommendation with parallel computation. *J. Intell. Fuzzy Syst.* **2019**, *36*, 4189–4199. [CrossRef]
38. Yang, J.; Leskovec, J. Overlapping Community Detection at Scale: A Nonnegative Matrix Factorization Approach. In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, Rome, Italy, 4–8 February 2013.
39. Newman, M.E.J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [CrossRef]
40. Zachary, W.W. An information flow model for conflict and fission in small groups. *J. Anthropolog. Res.* **1977**, *33*, 452–473. [CrossRef]
41. Krebs, V. (Unpublished). Available online: http://www.orgnet.com/ (accessed on 1 February 2015).
42. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef]
43. Wang, Q.; Fleury, E. Uncovering overlapping community structure. In *Complex Networks*; Springer: Berlin, Germany, 2011; pp. 176–186.