https://cke.um.ac.ir/ *DOI: 10.22067/cke.2022.69792.1007*

Multi-Agent Memetic Algorithm and its Application to Community Structure Detection in Complex Networks*

Research Article

Somayeh Ranjkesh[1] Behrooz Masoumi[2] Seyyed Mohsen Hashemi[3]

Abstract: A complex system is a system that has many components that are interdependent and appear as a whole and exhibit organized behavior. Community structure detection is an optimization problem in complex networks that involves searching for communities belonging to a network that shares nodes of a similar community with standard features. In this paper, we use a multi-agent memetic algorithm to detect the structure of the community in complex networks by optimizing the amount of modularity and calling it MAMA-Net. In the multi-agent memetic algorithm, agents are placed in a network-like environment to detect the community. Local search is used to find solution space. Having a local search in the memetic algorithm allows each member of the population to increase its evaluation function based on the suitability of its neighbors and achieve the desired result in minimum time. We compare the performance of MAMA-Net in detecting community structure with some standard algorithms. Both real-world and synthetic benchmark networks are used to evaluate the performance of the proposed method. The results show that MAMA-Net could detect communities more accurately than other comparable algorithms.

Keywords: Complex Systems, Multi-agent Systems, Memetic Algorithm, Community Detection

1. Introduction

A multi-agent system (MAS) is a type of system that has evolved from distributed artificial intelligence. A single agent cannot control the complexity of a structure; therefore a set of agents is used. Multi-agent systems have various features, including autonomy, distribution, self-organization, learning, and reasoning. One of the newest areas of application of multi-agent methods is complex systems such as networks and community structure detection [1]. Community structure is an essential topological feature of complex networks. Community detection is a challenging optimization problem involving searching communities belonging to a network or graph. The nodes of a community share standard features that allow detecting new features or working relationships in the network. Communities or clusters are usually groups of vertices, which are more likely to be connected than members of other groups. Still, the possibility of different patterns should not be ignored [2]. Examining the structure of communities is effective for studying the evolution of the entire network. Community detection algorithms provide a better understanding of complex network systems.

Complex systems are a new perspective on complex phenomena, and by having many connections between their elements, they exhibit different collective behaviors by studying every part of a complex system. In other words, a complex system represents a complexity paradigm that its constituent elements form a network with interoperable components and utilize holistic thinking. Economic systems, social systems (such as human networks), the Internet, and the brain are commonly known as complex systems. We can design complex networks with complex systems. The main reason for using complex networks is their flexibility and generality to express any natural structure. Complex networks that include nodes and nodes are represented by a diagram showing the relationship between nodes. Because of the large size of these networks, we should use algorithms with low computational costs to analyze the characteristics of these networks [3]. Community detection identifies groups on the web so that the communication within groups would be dense and confined between groups [4, 5]. Modularity is one of the most well-known criteria for measuring density within groups [6]. High modularity values indicate a significant difference between the partitions found and a random graph. Accordingly, we can consider the community detection problem a hybrid optimization problem by maximizing the modulus goal function [7].

One of the challenging problems in complex network analysis is detecting the community structure in large-scale networks. This study uses a multi-agent memetic algorithm to detect the structure of the community in complex networks by optimizing the amount of modularity and calling it MAMA-Net. Combining memetic algorithms (MA) and multi-agent systems increases the speed of convergence to achieve optimal global solutions. Multi-agent systems have proven to improve results on many optimization issues. This study investigates whether the combination of memetic algorithms and multi-agent systems can lead to more optimal answers to the problem of community detection than other existing methods. Local search is used to find solution space. Genetic operators generate new solutions in the global search process and the local search process discovers good quality solutions by searching around newly generated solutions. Local search in the memetic algorithm allows each member

* Manuscript received: April 04, 2021; Revised, November 9, 2021; Accepted: January 17, 2022.
[1]. Ph.D. Student, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran,
[2]. Corresponding author. Assistant Professor, Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran. **Email:** masoumi@qiau.ac.ir.
[3]. Assistant Professor, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.

of the population to increase the value of its evaluation function based on the suitability of its neighbors. We conclude that local search makes the algorithm more desirable in less time. The main contributions outlined in this study are as follows. We introduced the multi-agent memetic algorithm in complex networks using optimization of community structure detection. By achieving high modularity values, we achieved near-optimal solutions. Our proposed algorithm using a multi-agent system and local search can obtain good results from modular optimization to detect the structure of the community. In the proposed algorithm, we use the representation of locus-based adjacency, splitting and merging operators, hybrid crossover operators, and adaptive mutation operators. The tests on real-world and synthetic benchmark networks demonstrate the efficiency of our algorithm.

This section briefly discusses the importance and necessity of the work. The paper sections are as follows. Section 2 summarizes related works. Multi-agent systems and memetic algorithms are explained in the third section. The fourth section includes the explanation of the proposed algorithm, its operators, and its application to detect the community structure in complex explanation networks. The fifth section presents the proposed method's results, analysis, and comparison with other algorithms. Finally, the sixth section presents the conclusion of the study.

2. Review of related works

In recent years, mathematics, machine learning, statistics, and data mining have introduced various detecting communities in complex networks. Figure 1 shows the six main categories of these methods.

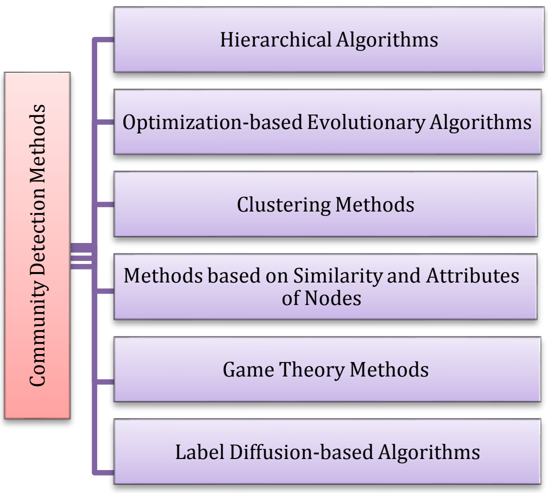


Figure 1. Classification of community detection methods in complex networks

The general philosophy of hierarchical algorithms is to find the edges between communities and eliminate them. If we remove all the edges between communities, those communities form separate components [8]. NM is a greedily condensing hierarchical algorithm [9]. In this algorithm, first, each node is a single community. Then, repetitively, the pairs of communities that lead to the maximum increase or the minimum decrease in modality are merged. The algorithm continues until all nodes are in a single community.

The main advantage of evolutionary optimization approaches is that they do not produce just one solution but a set of solutions. They divide the network into sections with different numbers of communities and analyze the structures of the communities [10]. Optimization-based algorithms can achieve optimal solutions and have the necessary efficiency on a large scale. GN algorithm is one of the most important primary and standard hierarchical algorithms for comparing community detection methods in complex networks [4]. This algorithm is defined based on centrality indicators for finding community boundaries that detect discovered communities' structures with high sensitivity and reliability. One of the most important evolutionary algorithms in optimization is the genetic algorithm. The genetic algorithm uses hybrid operators and special mutations to detect the community. Pizzuti [11] proposed a genetic algorithm for detecting communities in complex networks based on effective search operator and initialization, which expresses network flexibility and detail at different levels according to the analyst's needs. Integrating multi-agent systems with evolutionary algorithms is another example that solves constraints satisfaction and hybrid optimization problems with satisfactory results [12, 13]. There are also memetic algorithms and multi-agent for community detection, and neighborhood-based operators are designed and implemented. MA-Net is a memetic algorithm that uses modularity optimization to community detection. It is a fast and reliable algorithm for generating continuous solutions and uses an adjacency matrix for computations that use less memory [14]. MAGA-Net is a multi-agent genetic algorithm for optimizing modularity value for community detection. An agent represents a candidate solution. This algorithm can find the optimal world and can be used to solve large-scale networks [15]. One of the algorithms introduced in community detection is MIGA [16]. This algorithm uses modularity and community information to make it purposeful. This purposefulness of the algorithm causes a kind of stability and accuracy in the community detection. The meme-Net algorithm optimizes the modularity value to examine a network in different resolutions. Combining genetic algorithm and hill-climbing creates local search [17].

MLAMA-Net is an algorithm that defines a learning automaton for each node in the network with a chromosome and uses the interaction of evolutionary operators and local search to community detection [18]. There is a new genetic algorithm called GACD in which genetic representation reduces the search space, and the number of communities is determined automatically [19]. Pizzuti [20] proposed a multi-objective genetic algorithm to detect dense groups attached to nodes with scattered cross-links. Gong et al. [21, 22] proposed a multi-objective evolutionary algorithm for community detection by maximizing the density of internal degrees and reducing the density of external degrees. The synchronicity of these processes leads to the observation of structures with several resolutions. Ping et al. [23] proposed an algorithm called community MOCD-ACO. Each ant is responsible for solving a sub-problem in this algorithm. Ants are in different groups, and each group has a pheromone matrix. The operator simulates the local weight simulation annealing search so that the algorithm does not get caught in the local optimization and can find the near-optimal solution. The results of this algorithm show that mACOs have a high potential in solving community detection problems.

Clustering is one of the methods of community detection. Clustering means separating high-density areas in a set of characteristics from low-density areas. This clustering interpretation is very suitable for detecting communities. In clustering, it is always possible to extract clusters with the least inter-cluster distance and the most inter-cluster distance. Luo et al. [24] proposed an asymmetric NMF method via pointwise mutual information incorporated that is highly accurate. Lu et al. [25] used the NMF method to improve density peak clustering in community detection. Fiscarelli et al. [26] proposed a degenerate agglomerative hierarchical clustering algorithm to find the community structure that uses the reachability matrix. In this method, each vertex starts with its own cluster and, clusters are merged until merging is no longer possible. In [27], different clustering methods such as structural clustering algorithm for networks (SCAN), cluster based on structural feature (SA-cluster), community detection based on hierarchical clustering (CDHC) are introduced and evaluated. Chi et al. [28] proposed two frameworks for evolutionary spectrum clustering, including the parameters of maintaining cluster quality and membership. In evolutionary clustering, they must take two goals into account; that is, the result of good clustering should be well-proportioned with the current data and also not be significantly different from recent history at the same time.

Methods based on similarity and attributes of nodes consider a community group of similar nodes [29, 30]. Some local or global attributes compute the similarity between nodes. Each node belongs to the community to which the community nodes are most similar. Zhiwen et al. [31] proposed a new nonnegative matrix factorization (NMF) model for detecting network communities. This model has two parts, the community structure matrix and the node attribute matrix, which can detect internal connections between networks and determine the degree of network connection. The community structure matrix provides more information about the network by considering the relationships between nodes. This algorithm can effectively detect communities in real networks. Tang et al. [32] introduced a new criterion that considers structural similarities and features to identify society. Zhiwen et al. [33] introduced a new APT algorithm based on an integrated hybrid diffusion used for multiple learning in community detection.

We can use the idea of game theory to detect static communities, such as identifying guilds in multiplayer online games [34] and predicting trust between users on e-commerce sites [35]. Alvari et al. [36] proposed a new approach to identifying communities in dynamic social networks. It is a game theory approach to community detection in dynamic social networks. Each node acts as a logical representative that selects a set of predefined actions to maximize its profit performance.

In label diffusion-based algorithms, each node has a label. Then, each node expands its label step by step, covering more adjacent nodes. The algorithm performs this process until it reaches the community boundary. Xie et al. [37] proposed an algorithm called LabelRank. This algorithm is defined based on the distribution of labels in the network, which uses the node ID and publishes and ranks the labels in each node. Each node retains the labels it received from its neighbors. Nodes that have the same labels form a community. The LPA uses the idea of publishing labels to detect communities [38]. All nodes randomly update their labels in agreement with most of their neighbors. Node label updates can be synchronous or asynchronous. In sync synchronization, the label of a node in step t determines from its neighbor's label in step t. 1. Table 1 shows a summary of the methods mentioned above.

3. Multi-agent systems and memetic algorithms

A multi-agent system consists of agents to solve a problem and achieve the desired goal. Agents interact in multi-agent systems by communicating with each other and can work together in an environment, trying to accomplish a specific task and achieve a particular goal. Multi-agent systems provide the opportunity to calculate and optimize many complex problems.

Table 1. A brief summarization of community detectionmethods

Methods	Representative works
Hierarchical Algorithms	HCD [8], NM [9]
Optimization-based Evolutionary Algorithms	GN [4], GA-Net [11], MAEA-CmOP [12], MA-Net [14], MAGA-Net [15], MIGA [16], Meme-Net [17], MLAMA-Net [18], GACD [19], MOGA-Net [20], MOEA [22], MOCD-ACO [23]
Clustering Methods	SNMF [24], DPC [25], DAHCA [26], ACM [28]
Methods based on Similarity and Attributes of Nodes	SDP [29], CDASS [30], CDCN [31], NMNA [32], APT [33]
Game Theory Methods	SLM [35], D-Gt [36]
Label Diffusion-based Algorithms	LabelRank [37], LPA [38]

There are two critical issues involved in designing multi-agent systems: The first is the design of the agent, and the second is the environment's design for the performance and relationship between the agents. In agent designing, the key is the way of building an agent that is capable of performing independent tasks and autonomous actions. In designing a community or operating environment, the key is how to create agents that can interact with one another. This relationship implies cooperation, coordination, and negotiation between agents [39].

We can combine multi-agent systems and evolutionary algorithms to solve optimization problems. Agents live in a network, and we place each agent at a fixed point. All agents can increase their energy in competition with their neighbors and use domain knowledge. Combining evolutionary algorithms and multi-agent systems leads to convergence to optimal global solutions, which occurs at high speed. They are also used to solve large-scale problems with thousands of dimensions. This hybrid structure has been able to achieve good performance and reduce computational costs. In community detection, we define the agent as dividing a network and a candidate solution. Because agents live in a network-like environment, they are called network agents. They can exchange information with their neighbors. The memetic algorithm is the local genetic search or a hybrid genetic algorithm. A large part of the success of the memetic algorithm relies on the global convexity feature of the search space, and one of other advantages of using the memetic algorithm is reducing the area of probable solutions to a local optimum sub-space. The memetic algorithm is one of the evolutionary algorithms. According to their neighbors, each community member can increase its competency in this algorithm. The utility of each of the answers in this algorithm is calculated based on the evaluating function, and it generates new responses using the intersection and mutation operators. Finally, we can apply a local search to a set of solutions of that generation and a subset of the current generation (recent response sets, parents, new answers, and children). Generating new generations continues until fulfilling the stop condition. Memetic algorithms are hybrid optimization methods that add local search to the evolutionary optimization process, increase convergence speed, and solve complex optimization problems successfully. The local search strategy is the most critical key to the effectiveness of memetic algorithms. For this reason, these algorithms have been able to attract the interest of many researchers and are one of the most desirable methods of evolutionary optimization. Various problems use the memetic algorithm for optimizing their structure [40].

4. Multi-agent memetic algorithm in community detection

This study combines the memetic algorithm with a multi-agent system to use a new algorithm, MAMA-Net, in order to solve optimization problems. In this structure, agents live in a network-like environment. As we know, agents can live in the environment and apply activities according to what they understand, and specific goals can guide them. Each agent interacts with its environment, and other agents can increase its energy. We propose an agent as a candidate solution in the optimization problem whose energy value can be considered equal to the value of the evaluation function, and the goal is to maximize this amount of energy. Agents are fixed in the network-like structure and can only interact and exchange information with their neighbors. This process leads to disseminating information throughout the network structure, including agents. An agent can use local search to increase the value of its evaluation function based on the appropriateness of the importance of its neighbors' evaluation function. Each agent competes or cooperates with other neighboring agents to achieve its ultimate goal in optimization issues.

We have already mentioned that the combination of evolutionary algorithms and multi-agent systems has increased efficiency and improved results in solving optimization problems. In complex networks, if the nodes are easily grouped into a set of nodes so that each set of nodes are connected internally and densely, they are known as the community structure. We define a community as nodes with high internal interactions and relatively few external nodes in other groups. In community detection, the goal is to divide the data into groups that are divided into partitions $C = \{c_1, c_2, \ldots, c_r\}$ and are called communities, where r is the number of communities. Each c_a is a set of nodes. Some nodes are more interconnected than entire network nodes called communities in complex networks. Community detection shows the segmentation of the network and separates the communities from one graph [6].

The main goal in community detection is to find a part that can divide the network into the most meaningful communities. The mathematical definition of communities considers partition C of graph G to which node v belongs. The number of edges connecting node v to nodes belonging to partition C is equal tok_v^{in}, and the number of edges connecting node v to other nodes in the network isk_v^{ext}. If vertex v has only neighbors within,$k_v^{ext} = 0$, then partition C can be a good community for vertex $k_v^{in} = 0$.

After finding the communities, we need a criterion to show what quality the resulting association has, which is done by quality functions. Quality functions can be considered functions that consider a value as output for each community. Based on this output, we create the ability to evaluate the quality of a department. In community detection, most networks are considered complex. These networks represent a system or data that is not accidental, the source of which can be nature, community, or anything else. For example, consider the following complex network. This graph consists of 34 nodes and 78 edges, divided into two communities and shown in red and blue in Figure 2.

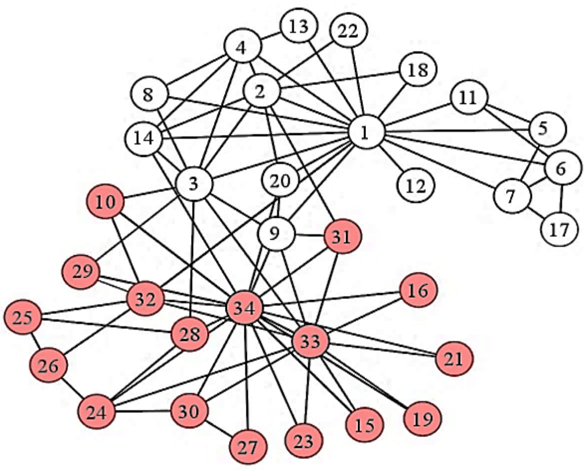


Figure 2. Complex network example

We use MAMA-Net to optimize community structure detection and describe its parameters in detail. We set the operators and parameters introduced in the algorithm structure here, which consists of six steps: neighborhood competition, crossover operator, mutation operator, self-learning operator, local search, and evaluation of individuals are defined. Figure 3 shows the parameters of our proposed algorithm. In the following, we define each of the operators used in our proposed method.

A. Neighborhood competition operator

We consider the agent in the a, b coordinates of the grid to be $Z_{a,b}$, and the agent with the highest energy among the neighbors to be $Max_{a,b}$. If the amount of energy $Z_{a,b}$ is greater than the amount of energy $Max_{a,b}$, then $Z_{a,b}$ will be the winner, and it is one of the best and can survive; otherwise, the agent $Z_{a,b}$ is replaced by $Max_{a,b}$. For the loser and replacement mode, two strategies can be proposed that emphasize exploitation or exploration that $Max_{a,b}$ chooses one of these strategies considering the probability. In the next section, where the application of the multi-agent memetic algorithm is in the problem of the community structure detection, we will explain the design of this operator in more detail. An agent locates at(a, b),$a, b = \{1, 2 \dots Size\}$ and an agent $Z_{a,b}$ consists of N genes. This paper applies the locus-based neighborhood display method to represent agents [41].

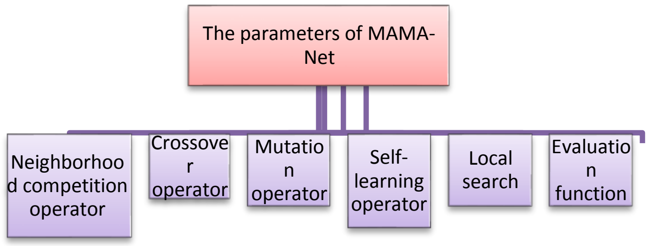


Figure 3. The main parameters of the proposed algorithm

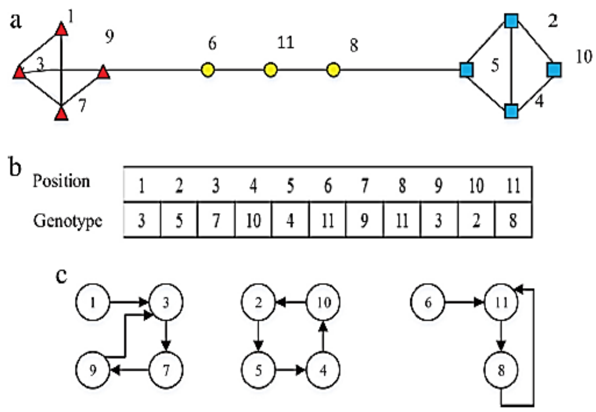


Figure 4. The locus-based representation of an individual

If there is a connection between nodes i and k, then the g_i gene takes the value of k. In this display $k(g_i = k)$ and the indices of i and k are in a similar community. We show the graph in Figure 4. This graph contains 11 nodes and 14 edges, and the relationships between nodes are displayed. For each node, we define a genotype that explores the relationship between specific nodes and communities. We see three areas, and the nodes 6, 8, and 11 are in one community.

In this method, we consider two approaches to detect the allele value of a random gene from each agent. The gene selected for each agent that is randomly determined is called the target gene. We choose an approach by choosing a random number. We compare the energy of the modified energy agent $(Z_{a,b})$ with the point of the best agent in the energy neighborhood $Max_{a,b}$, and if the energy improves, we save the new agent; otherwise, we will replace it with the best neighbor. If $Z_{a,b}$ satisfies Equation 1, it is a winner; otherwise, it is a loser.

$$\text{Energy}\ (Z_{a,b}) > \text{Energy}\ (\text{Max}_{a,b}) \tag{1}$$

We map $Max_{a,b}$ to [0, 1] and create a new agent $New_{a,b}$, and then $\text{New}_{a,b}$ is placed on the lattice-point. We used the division and integration operator to remove lower energy agents (smaller modularity values) from the network.

B. Crossover operator

We can use different crossover operators in optimization problems. New crossover operators have been proposed in [42] that generate unique individuals using the orthogonal design. We apply this operator to $Z_{a,b}$, and $Max_{a,b}$ to achieve the cooperation goal. Another type of crossover is the two-point type, in which there are two random points with the possibility of gene allocation between two points from $Max_{a,b,}$ [43]. The other crossover is uniform, which shifts each gene of $Z_{a,b}$, and $Max_{a,b}$ with a probability, and considers the newly created agent as $Z_{a,b}$. We use a combination of a two-point and uniform method in the crossover operator to apply the multi-agent memetic algorithm to the problem of the community structure detection, which we will explain in the next section. In the proposed algorithm, we use a two-point crossover operator. Two random points are selected, and the genes move together. We can use two-point, uniform integration with the flexibility of location-based neighborhood display. In the proposed method, we define a neighbor integration operator that consists of a combination of two types of two-point and uniform integration.

In this function, using two integration action strategies, an agent $Z_{a,b}$, is integrated with the best agent in its neighborhood $\max_{a,b}$. Each method is selected based on the P_x probability. If $u(0,1) < P_x$, the first strategy is adopted; otherwise, we will use the second strategy. The first strategy begins with the two-point integration and the random selection of points k_1 andk_2.

We replace the gene between the two selected points $\text{Max}_{a,b}$ with $Z_{a,b}$ if $u(0,1) < 0.5$; otherwise, the genes outside these selectable points. In the first strategy, we map the genes between positions $k1$ and $k2$ $Max_{a,b}$ to $Z_{a,b}$; otherwise, the rest of the genes *to* $Z_{a,b}$. In the second strategy, we use a uniform crossover operator in which $\text{Max}_{a,b}$, and $Z_{a,b}$ are merged, and $Z_{a,b}$ is replaced by a new agent.

C. Mutation operator

One of the mutation operators that we can use in optimization problems is the Gaussian mutation operator and changes a small part of $Z_{a,b}$. The Gaussian random number generator generates the new agent, and then $Z_{a,b}$ is replaced by $New_{a,b}$. Another type of mutation operator is the adaptive mutation. The number of genes changes to a neighbor's allele for each gene from one agent, taking into account a probability [29]. Depending on the type of optimization problem and its different applications, we can use one of these two mutation methods. Studies show that each method can achieve near-optimal solutions to maintain population diversity.

We use the adaptive mutation operator method to use the MAMA-Net algorithm in the problem of the community structure detection, which will explain the formula of the mutation method used in this application. The mutation operator provides random property and can escape local optimal points. It operates on a single sequence, with a small probability of mutating each bit of chromosomes.

The mutation is performed based on the probability P_b. If $u(0,1) < P_b$ then, the value of the gene is replaced by the neighbor allele's value in its neighborhood list. Moreover, in an adaptive mutation operator, the P_b changes to achieve better results [13]. We define the mutation operator as Equation 2.

$$P_{b'} = (\frac{t}{N_x} + 1)P_b \tag{2}$$

Variable N_x is the end criterion, and P_b is the mutation probability.

D. Self-learning operator

As mentioned earlier, agents can use their knowledge to understand their surroundings and perform activities according to the responses received from the environment. Most multi-agent systems seek to explore and monitor their surroundings. Each agent knows only the local environment around him .We assume that each agent provides information to other agents after becoming familiar with the environment. In such settings information sharing enables all stakeholders to know their environment and make better decision-making. The agents are selfish. In competitive environments, each greedy agent wants to maximize his usefulness by learning the behavior or weakness of the opponent. Learning in a collaborative environment encourages knowledge sharing and enhances knowledge in a competitive climate. Since agents are autonomous and only benefit them, competitive environments are more welcomed. In the proposed method, a network of small-scale sZ agents with the size $sZ_{size} \times sZ_{size}$ is created based on Equation 3.

$$sZ = \begin{cases} Z_{a,b} & a' = 1\,, b' = 1 \\ sZ_{a',b'} & otherwise \end{cases} \tag{3}$$

Variable $sZ_{a',b'}$ is created based on the neighborhood-based mutation operator on $Z_{a,b}$. We continue the splitting and merging operation to improve the operating energy until we have no improvement. The self-learning operator is essential in improving the performance of our proposed algorithm. Algorithm 1 shows this operator clearly. Figure 5 indicates the block diagram of the MAMA-Net.

Algorithm 1. Self-learning operator

Input:
$Z_{m,n}$: an agent in Z to do the self-learning operator;
Mutation: selfP_b;
selfZ_t: the agent lattice at the t the generation of selfZ;
selfB^t: the best agent in selfZ_0, selfZ_1,....., selfZ_t;
selfKB^t: the best agent in selfZ_t;
selfN_x: most generations created without improvement;
Output:
$Z_{m,n}$= selfB^t;
selfZ_0=define selfZ and update selfB^0;
while (i< selfN_x) do
t=t+1;
selfZ_t=apply the operators on selfZ_t;
Update selfKB^t;
if (E (selfKB^t)>E (selfKB^{t-1})) then
i=0;
selfB^t = selfKB^t;
else
i=i+1;
selfB^t= selfB^{t-1};
selfKB^t= selfB^t;
end
end

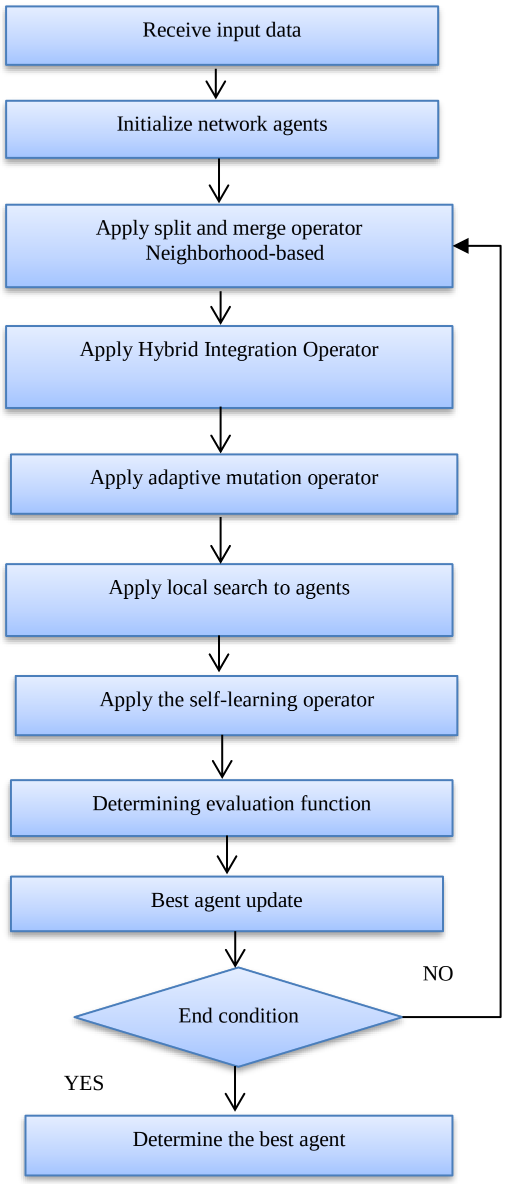


Figure 5. The block diagram of the MAMA-Net

E. Local search

The memetic algorithm is similar to the genetic algorithm. After each generation has improved, a local search for each chromosome occurs before being passed on to the next generation. By combining a global search with a local search, the memetic algorithms dramatically increase the global optimum's speed. This search aims to get a better chromosome value if there are more chromosomes in the neighborhood of a chromosome. In a multi-agent memetic algorithm in which chromosomes are involved in network structure, in local search, the fit of each agent is compared to four agents in its neighborhood. If there is an agent with higher merit, that agent takes the value of the neighboring agent. Local search causes the agent to increase the amount of energy according to the competence of its neighbors. Therefore, agents can move faster towards the optimal response and increase the degree of convergence. The following section, which uses the MAMA-Net for community structure detection, shows the neighborhood structure of local search agents. Local search can discover good quality solutions by searching around newly developed solutions. In the memetic algorithm, after generating a new generation and before replacing it with previous generations, the local search is applied to agents [44, 45]. Accordingly, for each agent, a neighborhood radius is considered that the neighbors of each agent $Z_{a,b}$ in the proposed method are determined based on Equation 4.

$$neighbors_{a,b} = \{Z_{a',b}, Z_{a,b'}, Z_{a,b"}, Z_{a",b}\} \tag{4}$$

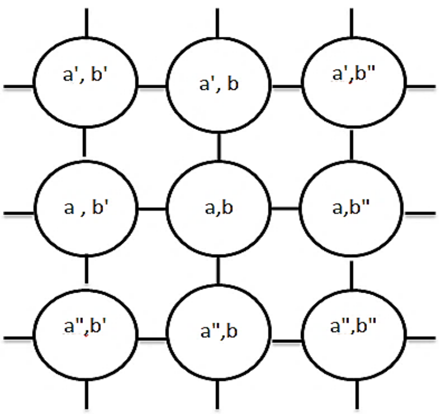
$$b' = \begin{cases} b-1 & b \neq 1 \\ Z_{size} & b=1 \end{cases}$$

$$a' = \begin{cases} a-1 & a \neq 1 \\ Z_{size} & a=1 \end{cases}$$

$$b" = \begin{cases} b+1 & b \neq Z_{size} \\ 1 & b = Z_{size} \end{cases}$$

$$a" = \begin{cases} a+1 & a \neq Z_{size} \\ 1 & a = Z_{size} \end{cases}$$

We compare each agent in terms of energy with neighboring agents. If we find an agent with higher energy than the original agent in the neighborhood (an agent), $Z_{a,b}$, we replace it with a higher energy agent. Figure 6 shows the neighborhood structure of an agent. Local search causes the algorithm to avoid the local optima [46].



Figre 6. The neighborhood structure of an agent

F. Evaluation Function

Depending on the type of optimization problem and its application, we can use different objective functions to evaluate individuals in the multi-agent memetic algorithm. Depending on the purpose of the problem, maximizing or minimizing the value of this function is considered. We can say that the evaluation function is equal to minimizing or maximizing the evaluation function of the solution. For example, in community structure detection, which we will explain in the next section, the goal is to find the structure of the community with maximum modularity. Modularity is one of the most commonly used criteria used in different methods. This criterion quantifies the clustering obtained from the whole graph and plays a significant role in determining the accuracy of the clustering. The higher the modularity obtained, the better the accuracy of the proposed method. The modularity criterion is the essential benchmark in evaluating community detection methods in complex networks. Modularity Q is a quality function that measures the quality of the network partitions. Equation 5 shows the modularity formula.

$$Q = \sum_{k=1}^{s}[\frac{h_k}{H} - (\frac{g_k}{2H})^2] \tag{5}$$

where H is the sum of all edges or connections in the network, s is the total number of communities, h_k is the number of edges within the community k, and g_k is the sum of the degrees of all nodes within the community k.

Normalized Mutual Information (NMI) is another criterion used to evaluate the results [47]. NMI determines the proximity of communities resulting from the proposed system to optimal communities. A and B are partitions of a network. If A and B are the same, $NMI(A,B) = 1$, and if A and B are entirely different, $NMI(A,B) = 0$ will be. Equation 6 shows $NMI\ (A,B)$.

$$NMI(A,B) = \frac{-2\sum_{i=1}^{k_A}\sum_{j=1}^{k_B} D_{ij} \log\left(\frac{D_{ij}N}{D_{i.}D_{.j}}\right)}{\sum_{i=1}^{k_A} D_{i.}\log\left(\frac{D_{i.}}{N}\right) + \sum_{j=1}^{k_B} D_{.j}\log\left(\frac{D_{.j}}{N}\right)} \tag{6}$$

Parameter k_A is the number of communities in A, k_B is the number of communities in B, D is the confusion matrix, and N is the number of elements. At the end of this section, we present the proposed MAMA-Net framework in Algorithm 2. We fully introduce the operators used in this algorithm and their functions.

Algorithm 2. MAMA-Net

```
Input:
  Crossover probability: Pa;
   Mutation probability: Pb;
   Ps: hybrid neighborhood operator;
   Ns: generations without improvement;
   Zt: the agent lattice;
   selfZ: the number of agents carried out self-learning
operator;
   Bt: the best agent in Z0, Z1,..., Zt;
   KBt[selfZ]: the best selfZ agents in Zt;
   KBt: the best agent in Zt;
   Output:
```

```
t= 0;
i= 0;
While (i< N_s) do
  t= t+1;
  Z_t= split and merging ;
  Z_t= hybrid neighborhood operator;
  Z_t= adaptive mutation operator;
  Z_t=apply a local search based on neighborhood
competition on Z_t by the best neighbor agent;
  KB^t[selfZ]:= finding the best selfZ agents in Z_t;
  for j= 1 to selfZ do
    if Learning (KB^t[j]= =True) then
      self-learning operator (Algorithm1)
    end
  end
  Update KB^t;
  if (E (KB^t) > E (KB^(t-1))) then
    i=0;
    B^t = KB^t;
  else
    i= i+1;
    B^t = B^(t-1)
KB^t = B^t;
  endend
```

5. Experimental results

In this section, we perform extensive comparative experiments to evaluate the effectiveness of our proposed MAMA-Net algorithm on two types of datasets. In this study, we use both real-world and synthetic networks to evaluate the performance of our proposed algorithm.

A. Real-world networks

We use four real-world networks: Karate [48], Dolphins [49], Polbook [50], and Football [4], as shown in Table 2. These are standard data used in almost all community detections[1].

Table 2. Dataset configurations

	Network	Nodes	Edges
1	Karate	34	78
2	Dolphins	62	159
3	Polbook	105	441
4	Football	115	613

B. Synthetic benchmark networks

Another dataset we use is large-scale synthetic LFR (Lancichinetti-Fortunato-Radicchi) networks with 1000 nodes [51], based on which we can evaluate the performance of our proposed method. We use the mixed parameter μ to control the structure of the community in the network that if the μ has values greater than 0.5, the structure of the community in the network will be uncertain. If this parameter has values less than 0.5, the structure of the community in the network will be significant and relatively straightforward. We use the NMI benchmark to evaluate the performance of MAMA-Net in this type of network. We run our algorithm 30 times in MATLAB 2016 on each dataset. We select solutions with the maximum value of NMI and modularity (Q) at each run.

We evaluate the performance of various community detection algorithms with modularity and Normalized Mutual Information (NMI). By modularity, we can determine the importance of the structure of the community. Normalized Mutual Information (NMI) evaluates the similarity between real communities and communities obtained by algorithms. As we said before, the evaluation criterion for our proposed method is modularity, which is between -1 and +1. If the modularity value is zero, we put all the nodes in a community, and if negative, it means a lot of error in finding communities.

We tested the proposed MAMA-Net algorithm on synthetic benchmark networks and real-world networks. The performances of the MAMA-Net algorithm were compared with several other community detection algorithms including CLACD [45], MLAMA-Net [14], CLA-Net [52], MIGA [16], GN (a greedy heuristic algorithm) [4], MAGA-Net [42], Meme-Net [17], MA-Net [14], MOCD-ACO [23], and CDCN [31]. These methods are different algorithms in community detection that use the modularity function and NMI criterion to evaluate the quality of discovered communities and have been able to obtain optimal results. We now compare our proposed algorithm with these methods to see if the multi-agent structure and local search can achieve better results than these algorithms. We will first briefly explain the idea of these algorithms. Cellular learning automata algorithms are new and accurate methods in detecting communities in complex networks. In this study, we used some proposed algorithms for comparison. The CLACD algorithm uses cellular automation to detect the structure of a community. This method can examine the global and local search space. The MLAMA-Net algorithm is a new community detection algorithm based on cellular learning automata in which some learning automata work together. The learning automata used in this algorithm interacts with both local and global environments, resulting in a more efficient structure of obtaining network communities.

The CLA-Net algorithm uses cellular automation to detect communities in complex networks. This algorithm models the entire network as irregular cellular learning automation (ICLA) and shows the optimal community structure through the evolution of cellular learning automata.

The MIGA algorithm uses genetic algorithms to optimize network modularity without knowing the number of initial communities. This algorithm is also used for extensive networks and discards a new community variance criterion. MIGA is an improved genetic algorithm to approach the largest modularity. Genetic-based algorithms may be locally optimized and not suitable for large-scale networks. MAGA-Net is a multi-agent genetic algorithm for optimizing modularity value for community detection. An agent represents a candidate solution. This algorithm can find the optimal world and can be used to solve large-scale networks. We can improve the results by adding local search to the genetic algorithm and considering the neighbors'

[1] http://www-personal.umich.edu/~mejn/netdata/

competencies. Some methods based on memetic algorithms for community detection have been proposed, which we used two essential models used in most studies for comparison. The application of the Meme-Net algorithm is the optimization of the modularity density function in detecting the structure of the community and examining different network resolutions. This algorithm uses a combination of genetic algorithms and mountaineering strategy as a local search method.

The MA-Net algorithm is a way to detect communities in a network by optimizing the modularity value, which is fast and reliable. This algorithm can continuously discover reasonable solutions to the community detection problem with a small deviation from the near-optimal solution for modularity optimization. We compare our proposed method with these algorithms to check whether the multi-agent structure in the memetic algorithm has achieved better results than the methods introduced above. The value of parameters used in our proposed algorithm is according to Table 3.

Table 3. Parameter value

$selfN_x$	N_x	$self_{-pub}$	P_x	P_b	P_a	self Z_{size}	Z_{size}
50	10	0.02	o.5	0.05	0.6	3	5

Table 4 shows the results of comparing our proposed MAMA-Net algorithm with other algorithms in four real-world networks. The results of Table 4 show that our proposed algorithm has increased and improved the average modularity value compared to other algorithms.

A comparison of values shows that the results obtained with MAMA-Net are better than other algorithms. The proposed algorithm can improve the results with less time and more speed. Moreover, the maximum modularity values obtained from the proposed algorithm achieved better results in all four networks. Our proposed algorithm obtained the value 0.4342 for the maximum modularity in the Karate network. In this network, CDCN was able to rank second with a value of 0.4253 for maximum modularity.

MAGA-Net and MOCD-ACO have been able to obtain almost identical answers. Comparing the results, we see that GN has a lower value than others. The values obtained by our proposed algorithm and the CDCN method in the Football network for the maximum modularity are 0.6119 and 0.6074, respectively, which have reached a better result than other algorithms. The maximum modularity values in the Dolphins' network with the numbers 0.5341 and 0.5286 are related to our proposed algorithm and MAGA-Net, respectively. The lowest value obtained at 0.4955 belongs to CLANet.

Comparing the results from the Polbook network, we conclude that our proposed method and the CDCN have achieved more desirable values for modularity. MAGA-Net is in third place, and GN is at the bottom of our ranking to get the maximum modularity value. Examining the results obtained on real-world networks, we can say that the multi-agent structure and local search have improved the memetic algorithm and increased the evaluation function. Moreover, the community structure matrix and the node attribute matrix in the CDCN method have shown good efficiency in modularity calculation. Figure 7 shows the maximum modularity value obtained in our proposed method and other algorithms on real-world networks.

Table 4. Comparison of the MAMA-Net with the other algorithms in terms of modularity Q

Algorithm	Modularity	Dolphins	Karate	Football	Polbook
MAMA-Net	Q_{max}	0.5341	0.4342	0.6119	0.5308
MAGA-Net	Q_{max}	0.5286	0.4198	0.6046	0.5273
MLAMANet	Q_{max}	0.5277	0.4184	0.6050	0.5272
CLACD	Q_{max}	0.5277	0.4132	0.6044	0.5223
CLANet	Q_{max}	0.4955	0.4104	0.6046	0.5181
MIGA	Q_{max}	0.5210	0.4128	0.5911	0.5272
MA-NET	Q_{max}	0.529	0.4201	0.6052	0.527
GN	Q_{max}	0.519	0.4010	0.599	0.510
MemeNet	Q_{max}	0.518	0.4023	0.601	0.523
MOCD-ACO	Q_{max}	0.5258	0.4197	0.5999	0.5208
CDCN	Q_{max}	0.5266	0.4253	0.6074	0.5289

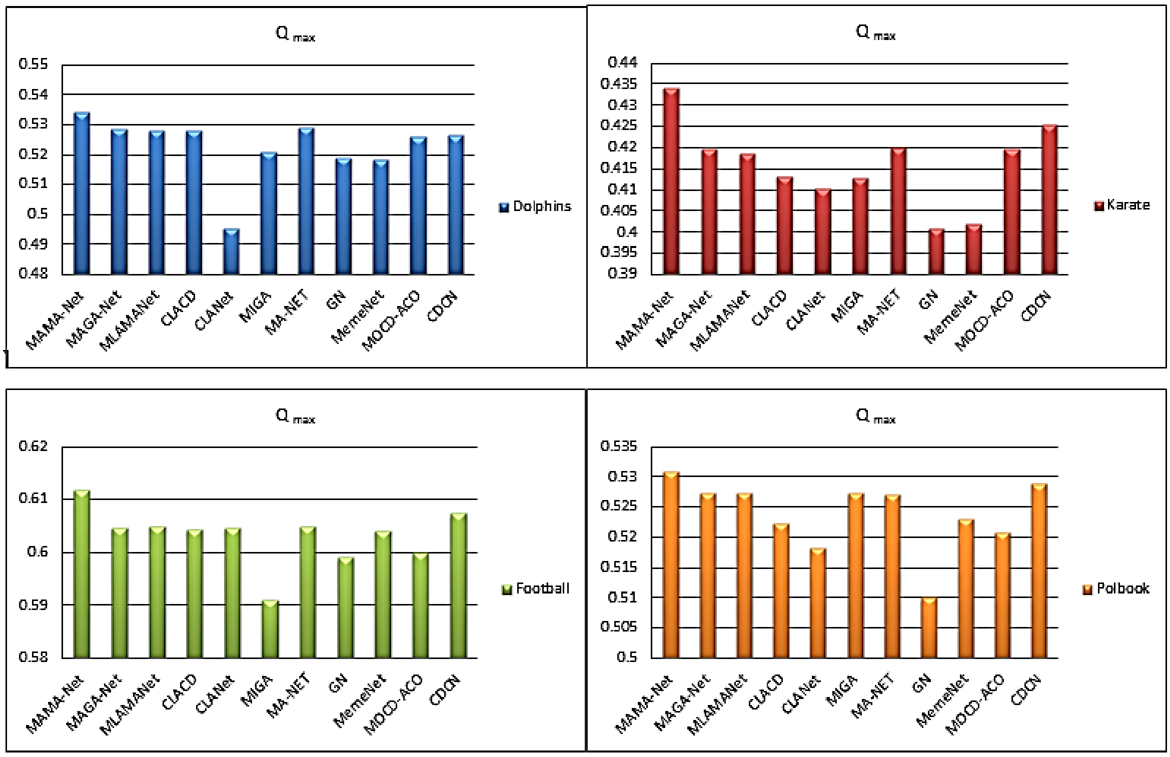


Figure 7. Compare MAMA-Net with other $Qmax$ algorithms on real-world networks

Based on the results in Table 4, we can say that the multi-agent structure in our proposed method and the learning method proposed optimized the modularity value better than other algorithms in this domain. We see improvement in the results of our proposed algorithm compared to other algorithms. Table 5 shows the execution time of MAMA-Net to calculate the maximum modularity value on four real-world networks.

Table 5. The execution time of MAMA-Net

MAMA-Net		
Network	Times	Q_{max}
Karate	1.1258	0.4342
Dolphin	1.4088	0.5341
Polbook	1.2908	0.5308
Football	4.9529	0.6119

This section evaluates NMI values in the Karate, Football, Polbook, and Dolphins datasets and compares our method with other algorithms. We use NMI to assess the partitioning results of all methods. We obtain the NMI value for MAMA-Net, which estimates the similarity between the detected and the actual partitions. Figure 8 shows the results. The results show that the proposed algorithm (MAMA-Net) achieved better results by running on all four networks.

Our proposed algorithm obtained the value 0.9127 for the NMI in the Karate network. CDCN and MOCD-ACO obtained 0.9047 and 0.9024 in the NMI calculation, respectively. Our highest comparison rank is the NMI obtained in the Dolphins' network in MAMA-Net and MAGA-Net with 0.7942 and 0.7654. The values obtained for other algorithms are 0.6324, 0.6431, 0.6523, 0.6923, 0.7611, 0.7605, 0.7645, 0.7614, and 0.7403. In the Polbook network, the MAMA-Net, MLAMANet, MAGA-Net, and CLACD algorithms obtained 0.8207, 0.7941, 0.7921, and 0.7915 for NMI, respectively. Again, we see that the most optimal result belongs to our proposed algorithm. The NMI value obtained in our proposed algorithm in the Football network is 0.9511, which is the best result compared to other methods.

MAGA-Net and CDCN with 0.9102 and 0.9089 were able to occupy the next ranks in comparison. The results show that MAMA-Net performs better than all the comparable algorithms for community detection. In the Karate, Dolphins, Football, and Polbook datasets, NMI obtains the values 0.9127, 0.7942, 0.9511, and 0.8207, respectively, which are better than other methods. The NMI value is improved using our proposed algorithm. Our results show that the partitions obtained by the MAMA-Net are more similar to the real ones. Figure 8 shows the NMI comparison chart obtained by MAMA-Net and other algorithms.

We also apply the MAMA-Net algorithm on LFR benchmark networks and other community detection algorithms. The network size is 1000, and the node degree is between 0 and 50. Since some algorithms can perform well even when the community structure in LFR benchmark networks is indistinct, the mixing parameter μ varies from 0.1 to 0.5. The NMI is used to evaluate the performances of different algorithms. Figure 9 shows the results.

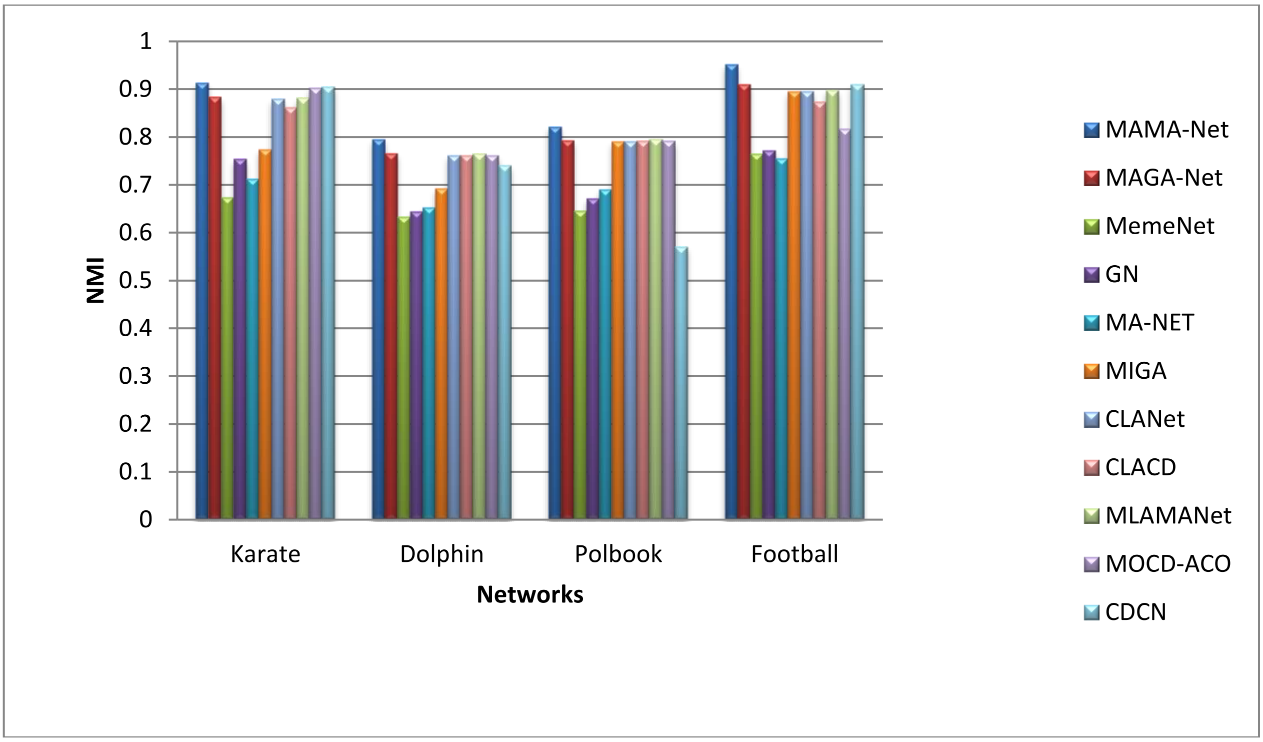


Figure 8. The average NMI for MAMA-Net and other algorithms

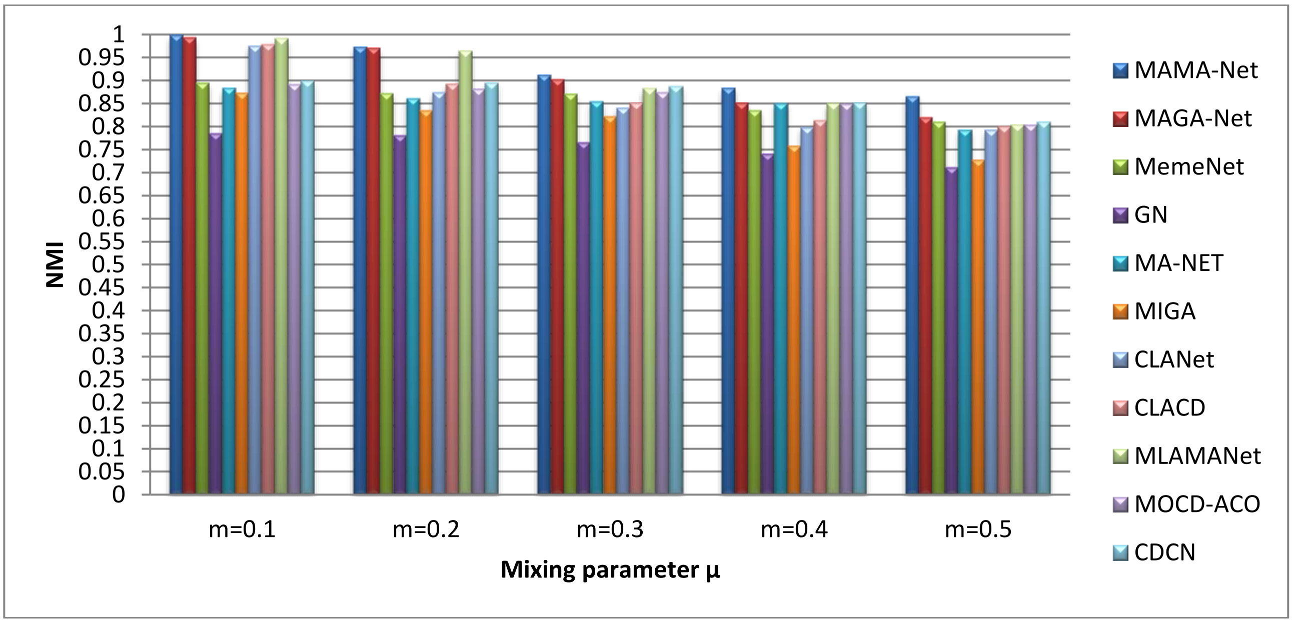


Figure 9. Values obtained for the average NMI by MAMA-Net and other algorithms in standard LFR networks

When $\mu \leq 0.5$, our MAMA-Net algorithm can always find the correct partitions of the benchmark networks. According to Figure 9, our proposed method has always achieved the highest NMI value. The NMI values in our proposed algorithm are 0.9984, 0.9727, 0.9116, 0.8839 and 0.8654 for $\mu = 0.1 \text{to} \mu = 0.5$, respectively. We observe that by increasing the mixing parameter μ, the performance of most algorithms decreases. However, our proposed algorithm still performs better than other algorithms and has the highest NMI value. That is, the communities discovered by our proposed algorithm are more accurate. When $\mu = 0.5$, the NMI values obtained from the GN, MIGA, and CLANet algorithms are 0.7111, 0.7265, and 0.7924, respectively, which are lower than other algorithms. The algorithms cannot detect the community structure correctly as the NMI value decreases and their accuracy decreases.

C. Statistical analysis

In this study, we used two independent samples of T-test on four criteria of Dolphins, Football, Karate, and Polbook to evaluate whether the answers obtained by the proposed algorithm have improved over MAGA-Net independent of quantitative data. We chose MAGA-Net for comparison because it has a multi-agent structure, and we examine whether our proposed method of detecting communities has performed better than it. The t-test is a distribution or, in fact, a set of distributions and its application is to test unknown hypotheses in the community. The significance of this distribution is that it enables the researcher to obtain information about the community in smaller samples, at least two individuals. The T-test assumes that each model has its distribution, determined by calculating the degrees of freedom. The T-test was applied to find a significant difference between the two populations. Here, the mean of two populations is the results of the proposed method and MAGA-Net. Greater values (positive or negative) of T denote the higher probability of rejecting the null hypothesis. Closer values of T to zero indicate a higher likelihood of confirming the null hypothesis. Table 5 compares Q values in MAMA-Net and MAGA-Net in 20 replicates. Here we show the output of the T-test algorithm in Minitab software. We perform the test at a 95% confidence level with 0.05 alpha. As shown in Table 6, the T-values are very far from zero. $C1$ means the proposed method's data, and $C2$ means the compared method's data. Equations 8 and 9 introduce the parameters used in the T-test.

μ_1: mean of $C2$

μ_2: mean of $C1$

Difference: $\mu_1 - \mu_2$

(8)

Null hypothesis H_0: $\mu_1 - \mu_2 = 0.05$

Alternative hypothesis H_1: $\mu_1 - \mu_2 \neq 0.05$

(9)

Table 6. Result of T-test on four benchmarks

Network	Dolphin	Football	Karate	Polbook
P-value	0.0000	0.0000	0.0000	0.0000
T-value	-4.40	-4.26	-4.65	-4.22

The results in Table 6 show that we obtain T-value = -4.40 and P-value = 0.0000 in the Dolphin network and indicate that we reject the null hypothesis. The differences in the mean data of the proposed algorithm and the compared algorithm are significant. We also get similar results for other three networks. The null hypothesis ($H0$) means that the variance of the two data sets compared is equal, and the alternative hypothesis ($H1$) means that the two data groups' variance is not equal. The greater the value of T than 0.05, the more likely it is to reject the null hypothesis. The smaller the value of T and closer to zero, the more likely it is to accept the null hypothesis; there is no significant difference between the means of the groups compared ($C1$ and $C2$). We show the box diagrams for comparing MAMA-Net and MAGA-Net for four networks in Figures 10 to 13. These graphs show that the proposed algorithm performs better than the MAGA-Net. Table 6 shows the descriptive statistics and estimation of differences for the four networks. In Table 7, N is the number of iterations of the algorithms and StDev is the standard deviation, and SE is the standard error. By standard deviation, the root of the sample variance is an index that calculates how the sample data is distributed and changed around its mean and the standard error in the case of multiple (n) population sampling by placing the same volume. After n repetitions of the statistical method (for example, the average), we get a new sample with volume n. We should also mention the degree of freedom (DF) for the difference in means.

The degree of freedom is the value we use to indicate the sample size or samples used in a statistical test. The method of reporting the degree of freedom is different in all types of statistical tests. Before the significance of the test result is checked, the degree of freedom should be calculated accurately and correctly. The Minitab software calculates the degree of freedom automatically.

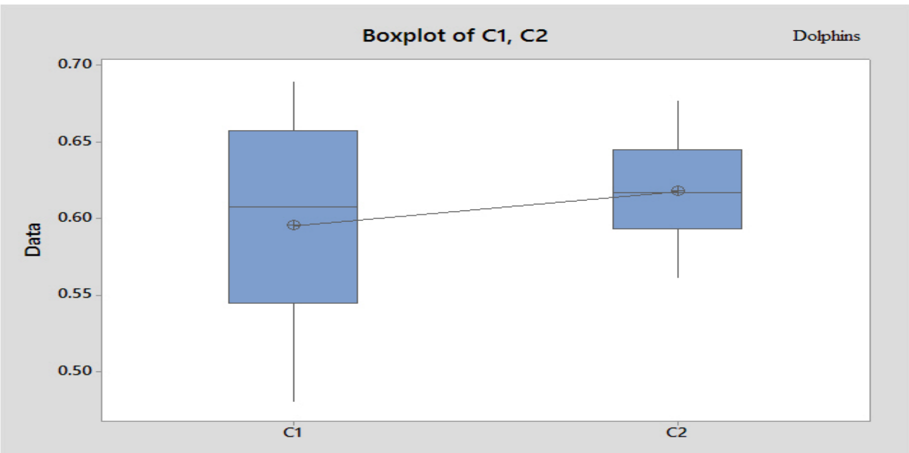


Figure 10. The box diagrams of MAMA-Net and MAGA-Net for the Dolphins network

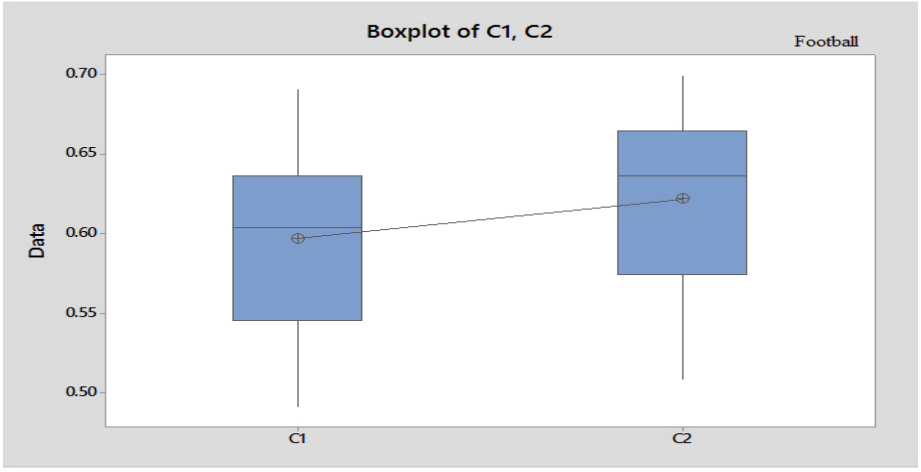


Figure 11. The box diagram of MAMA-Net and MAGA-Net for the Football network

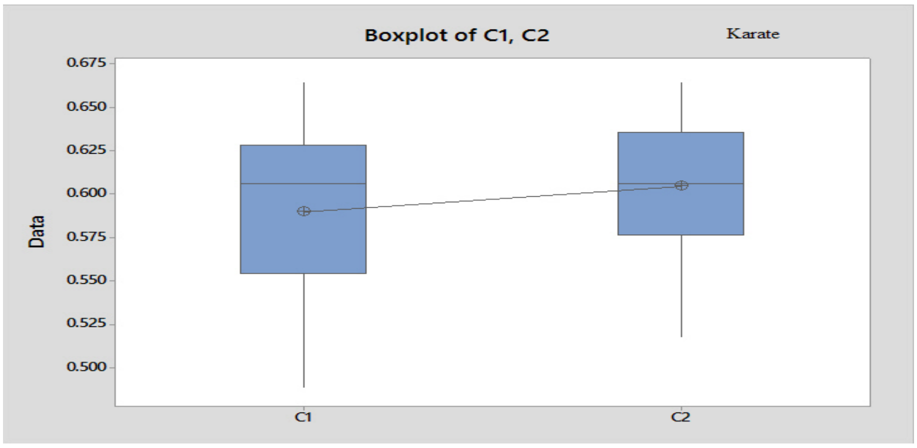


Figure 12. The box diagram of MAMA-Net and MAGA-Net for the Karate network

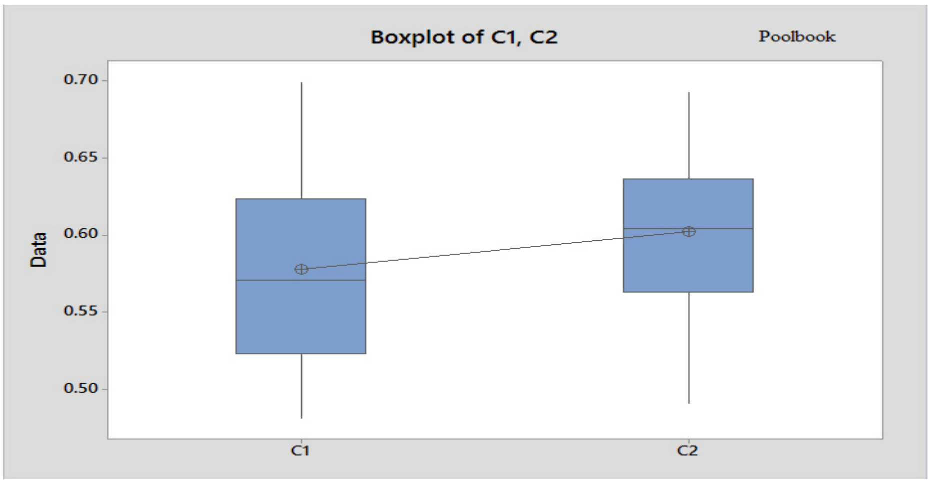


Figure 13. The box diagrams of the MAMA-Net and the MAGA-Net for the Polbook network

Table 7. The results of descriptive statistics and estimation for difference for four benchmarks

Network	Sample	N	Mean	StDev	SE Mean	Difference	90% CI for Difference	T-Value	DF	P-Value
Dolphins	C1	20	0.5953	0.0656	0.015	-0.0224	(-0.0504, 0.0056)	- 4.40	28	0.000
	C2	20	0.6177	0.0333	0.0074					
Football	C1	20	0.5972	0.0540	0.012	-0.0249	(-0.0545, 0.0047)	-4.26	37	0.000
	C2	20	0.6221	0.0570	0.013					
Karate	C1	20	0.5898	0.0500	0.011	-0.0147	(-0.0382, 0.0088)	-4.65	35	0.000
	C2	20	0.6045	0.0370	0.0083					
Polbook	C1	20	0.5779	0.0624	0.014	-0.0246	(-0.0544, 0.0053)	-4.22	35	0.000
	C2	20	0.6024	0.0486	0.011					

According to the results of Table 5, we see that the values of T-value = -4.26, P-value = 0.0000, and DF = 37 are obtained for the Football network and indicate the rejection of the null hypothesis. Also for the kKarate network the values of T-value = -4.65, P-value = 0.0000, and DF = 375 and for the Polbook network the values of T-value = -4.22, P-value = 0.0000, and DF = 35 have been calculated. We can conclude that our proposed algorithm detects community structures more accurately and performs better than the comparison method.

6. Conclusion

A complex system is a system whose members are interdependent and appear as a single whole. These systems have many components and exhibit organized behavior. We can model complex systems as complex networks. The main reason for using complex networks is their flexibility and generality to express any natural structure. Community detection is an optimization problem in complex networks that involves searching for communities belonging to a network that shares the nodes of a similar community with standard features that identify new features or relationships in the network. By combining a global search with a local search, the memetic algorithms dramatically increase the global optimum's speed. We use local search to find solution space. Genetic operators generate new solutions in the global search process, and the local search process discovers good quality solutions by searching around newly generated solutions. Local search can discover good quality solutions by searching around newly developed solutions.

Therefore, this study provides a multi-agent memetic algorithm and then uses this proposed algorithm, MAMA-Net, in complex networks for community structure detection by optimizing modularity value. In the multi-agent memetic algorithm, the agents are located in a lattice-like environment to detect the community. The local search in the memetic algorithm allows the members of the population to increase their level of competence compared to their neighbors in less time and achieve the desired result. We investigated the application of this algorithm for community structure detection, and compared the performance of the MAMA-Net with some well-known algorithms. The results showed that the MAMA-Net algorithm achieves excellent and improved solutions. Both in the modularity criterion and the NMI criterion, MAMA-Net has achieved near-optimal solutions. Our proposed algorithm performs better than other methods and achieves better results in less time. In addition, our experiments have shown that the MAMA-Net approach can always discover good solutions for the community detection problem for modularity optimization.

7. References

[1] Guo, L., Nian, X., Pan, H., "Leader-following consensus of multi-agent system with diffusion", *Asian J. Control*, vol.16 (1), pp. 188–197, (2014).

[2] Ye, Z., Hu, S., Yu, J., "Adaptive clustering algorithm for community detection in complex networks", *Phys. Rev*. vol. 78(4), (2008).

[3] Fortunato, S., Castellano, C., "Community structure in graphs", *Computational Complexity*. Springer; pp. 490–512, (2012).

[4] Girvan, M., Newman, M., "Community structure in social and biological networks", Proc. Natl. Acad. Sci. vol. 99(12), pp. 7821–7826, (2002).

[5] Newman, M., Girvan, M., "Finding and evaluating community structure in networks", *Phys. Rev*. vol. 69 (2), pp. 026113, (2004). https://doi.org/10.1103/PhysRevE.69.026113.

[6] Newman, M., "Modularity and community structure in networks", Proc. Natl. Acad. Sci, vol. 103(23), pp.8577–8582, (2006).

[7] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D., "On modularity clustering", *IEEE Trans*. Knowl. Data Eng. Vol. 20 (2), pp. 172–188. https://doi.org/10.1109/TKDE.2007.190689, (2008).

[8] Li, T., Lei, L., Bhattacharyya Sh., Berge, K., Sarkar, P., Bickel, P., Levina, E., "Hierarchical Community Detection by Recursive Partitioning", *Journal of the American Statistical Association*, Published online, (2020). https://doi.org/10.1080/01621459.2020.1833888.

[9] Ahuja, M., "Problem domains in complex networks", *IOSR Journal of Computer Engineering (IOSR-JCE).* vol. 18(5), pp. 65-68, (2016).

[10] Rahimi, S., Abdollahpouri, A., Moradi, P., "A multi-objective particle swarm optimization algorithm for community detection in complex networks", *Swarm and Evolutionary Computation*, pp. 297-309, (2018). https://doi.org/10.1016/j.swevo.2017.10.009.

[11] Pizzuti, C., "Ga-net: A genetic algorithm for community detection in social networks", *Parallel Problem Solving from Nature*–PPSN X, Springer; pp. 1081–1090. https://doi.org/10.1007/978-3-540-87700-4_107, (2008).

[12] Liu, J., Zhong, W., Jiao, L., "A multi-agent evolutionary algorithm for combinatorial optimization problems", *IEEE Trans. Syst. Man Cybern*, vol. 40 (1), pp. 229–240. https://doi.org/10.1109/TSMCB.2009.2025775, (2010).

[13] Liu, J., Zhong, W., Jiao, L., "A multi-agent evolutionary algorithm for constraint satisfaction problems", *IEEE Trans. Syst. Man Cybern.* vol. 36 (1), pp. 54–73, (2006). https://doi.org/10.1109/TSMCB.2005.852980.

[14] Naeni L., Berretta, R., Moscato, P., "MA-Net: A reliable memetic algorithm for community detection by modularity optimization", *In Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*. Springer, pp. 311–323, (2015). https://doi.org/10.1007/978-3-319-13359-1_25.

[15] Li, Z., Liu, J., "A multi-agent genetic algorithm for community detection in complex networks", *Physica* A., vol. 449, pp. 336–347, (2016). https://doi.org/10.1016/j.physa.2015.12.126.

[16] Shang, R., Bai, J., Jiao, L., Jin, C., "Community detection based on modularity and an improved genetic algorithm", *Phys. Stat. Mech. Appl*, vol. 39 (2), pp. 1215–1231, (2013). https://doi.org/10.1016/j.physa.2012.11.003.

[17] Gach, O., Hao, J., "A Memetic Algorithm for Community Detection in Complex Networks", *Lecture Notes in Computer Science,* pp. 327-336, (2012). https://doi.org/10.1007/978-3-642-32964-7_33.

[18] Mirsaleh, M, Meybodi, M., "A Michigan memetic algorithm for solving the community detection problem in a complex network", *Neurocomputing*, vol. 21(4), pp. 535–545, (2016). https://doi.org/10.1016/j.neucom.2016.06.030

[19] Shi, C., Wang, Y., Wu, B., Zhong, C., "A new genetic algorithm for community detection", in *International Conference on Complex Sciences*, vol. 1298–1309, (2009).

[20] Pizzuti, C., "A multi-objective genetic algorithm to find communities in complex networks", *IEEE Trans. Evol. Comput*. vol. 16(3), pp. 418–430, (2012).

[21] Gong, M., Ma, L., Zhang, Q., Jiao, L., "Community detection in networks using a multi-objective evolutionary algorithm with decomposition", *Physica A*, vol. 391(15), pp. 4050–4060, (2012).

[22] Gong, M., Chen, X., Ma, L., Zhang, Q., Jiao, L., "Identification of multi-resolution network structures with a multi-objective immune algorithm", *Appl. Soft Comput*. vol, 13(4), pp. 1705–1717, (2013).

[23] Ji Ping, Zhang Shanxin, Zhou ZhiPing. "A decomposition-based ant colony optimization algorithm for the multi-objective community detection", *Journal of Ambient Intelligence and Humanized Computing,* vol. 11, pp. 173–188, (2020). https://doi.org/10.1007/s12652-019-01241-1.

[24] Luo, X., Liu, Z., Shang, M., Lou, J., Zhou, M., "Highly accurate community detection via pointwise mutual information incorporated symmetric nonnegative matrix factorization", *IEEE Transactions on Network Science and Engineering,* vol. 8 (1), pp. 463–476, (2021).

[25] Lu, H., Sang, X., Zhao, Q., and Lu, J., "Community detection algorithm based on nonnegative matrix factorization and improved density peak clustering", *IEEE Access,* vol. 8, pp. 5749–5759, (2020).

[26] Fiscarelli, A., Brust, M., Danny, G., Bouvry, P., "A vertex-similarity clustering algorithm for community detection", *Journal of Information and Telecommunication*, vol. 4(1), (2020). https://doi.org/10.1080/24751839.2019.1686683.

[27] Agrawal, S., Patel, A., "Clustering Algorithm for Community Detection in Complex Network: A Comprehensive Review", *Computer Science*, (2020). https://doi.org/10.2174/2213275912666190710183635

[28] Chi, Y., Song, X., Zhou, D., Hino, K., and Tseng, B., "ACM Trans", Knowl. Discov. Data 3, vol. 17(1), (2009).

[29] Tang, F., Wang, Ch., Su, J., Wang, Y., "Semidefinite programming based community detection for node-attributed and multiplex networks", *Communications in Statistics - Simulation and Computation*, Published online. (2020). https://doi.org/10.1080/03610918.2020.1847291.

[30] Zarandi, F., Rafsanjani, M., "Community detection in complex networks using structural similarity", *Physica A: Statistical Mechanics and its Applications,* vol. 503, pp. 882-891, (2018). https://doi.org/10.1016/j.physa.2018.02.212.

[31] Ye Zhiwen, Zhang Hui, Feng Libo, and Shan Zhangming. CDCN: A New NMF-Based Community Detection Method with Community Structures and Node Attributes. Wireless Communications and Mobile Computing, (2021). https://doi.org/10.1155/2021/5517204.

[32] Tang, F., Ding, W., "Community detection with structural and attribute similarities", *Journal of Statistical Computation and Simulation*, vol. 89(4), (2019). https://doi.org/10.1080/00949655.2019.1568435.

[33] Chen, Q., Qiao, T. L., Hu, F., Li, Y., "Community detection in complex network based on the APT method", *Pattern Recognition,* (2020). https://doi.org/10.1016/j.patrec.2020.07.021.

[34] Alvari, H., Lakkaraju, K., Sukthankar, G., and Whetzel, J., "Predicting guild membership in massively multiplayer online games", *In Proceedings of the International Conference on Social Computing,*

Behavioral-Cultural Modeling, and Prediction, Washington, pp. 215–222, (2014).

[35] Beigi, G., Alvari, H., Jalili, M., and Sukthankar, G., "Leveraging community detection for accurate trust prediction", In ASE International Conference on Social Computing, Palo Alto, CA, May, (2014).

[36] Alvari Hamidreza, Hajibagheri Alireza, Sukthankar Gita. Community detection in dynamic social networks: A game-theoretic approach. In IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, (2014).

[37] Xie, J., and Szymanski, B., "Labelrank: A stabilized label propagation algorithm for network community detection", In-Network Science Workshop (NSW), IEEE, pp. 138-143, (2013).

[38] Zhang, X., Ren, J., Song, Ch., and Zhang, Q., "Label propagation algorithm for community detection based on node importance and label influence", *Physics Letters A*, vol. 381(33), pp. 2691-2698, (2017). https://doi.org/10.1016/j.physleta.2017.06.018.

[39] Zhong, W., Liu, J., Xue, M., Jiao, L., "A multi-agent genetic algorithm for global numerical optimization", *IEEE Trans*, vol. 34 (2), pp. 1128–1141, (2004). https://doi.org/10.1109/TSMCB.2003.821456.

[40] Maesani, A., Lacca, G., Floreano, D., "Memetic Viability Evolution for Constrained Optimization", *IEEE Transactions on Evolutionary Computation*. vol. 1(20), pp. 125-144, (2016).

[41] Park, Y., Song, M., "A genetic algorithm for clustering problems", *In Proceedings of the Third Annual Conference on Genetic Programming*, (1998).

[42] Leung, Y., Wang, Y., "An orthogonal genetic algorithm with quantization for global numerical optimization", *IEEE Trans. Evol. Comput.* pp. 41–53, (2001).

[43] Li, Zh., Liu, J., "A multi-agent genetic algorithm for community detection in complex networks", *Physica A*, vol. 449, pp. 336–347, (2016). https://doi.org/10.1016/j.physa.2015.12.126.

[44] Liu, X., Murata, T., "Advanced modularity specialized label propagation algorithm for detecting communities in networks", *Physica A. Statistical Mechanics and its Applications*, vol. 398(7), (2010). https://doi.org/10.1016/j.physa.2009.12.019.

[45] Rotta, R., Noack, A., "Multilevel local search algorithms for modularity clustering", *J. Exp. Algorithmic*, vol. 6(2), pp. 1594-1605, (2011). https://doi.org/10.1145/1963190.1970376.

[46] Liu, J., Liu, T., "Detecting community structure in complex networks using simulated annealing with - means algorithms", *Physica A: Statistical Mechanics and its Applications*, vol. 389(11), pp. 2300-2309, (2010). https://doi.org/10.1016/j.physa.2010.01.042.

[47] Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A., "Comparing community structure identification", *J. Stat. Mech. Theory Exp*. (2005). https://doi.org/10.1088/1742-5468/2005/09/P09008.

[48] Zachary, W., "An information flow model for conflict and fission in small groups", *J. Anthropol*. Res. (1977). https://doi.org/10.1086/jar.33.4.3629752.

[49] Ma, L., "Multi-level learning-based memetic algorithm for community detection", *Applied Soft Computing*, vol. 19(0), pp. 121-133, (2014). https://doi.org/10.1016/j.asoc.2014.02.003

[50] Krebs, V., "A network of books about US politics", (2008). http://www.orgnet.com/.

[51] Lancichinetti, A., Fortunato, S., Radicchi, F., "Benchmark graphs for testing community detection algorithms", Phys. Rev. E, (2008).

[52] Khomami, M., Rezvanian, A., Meybodi, M., "A New Cellular Learning Automata-based Algorithm for Community Detection in Complex Social Networks", *Journal of Computational Science*, vol. 24, pp. 413-426, (2018). https://doi.org/10.1016/j.jocs.2017.10.009.