Community Detection Algorithm Based on Artificial Fish Swarm Optimization

Eslam Ali Hassan[1,3], Ahmed Ibrahem Hafez[2,3], Aboul Ella Hassanien[1,3], and Aly A. Fahmy[1]

[1] Faculty of Computers and Information, Cairo University, Egypt
eslam.ali@fci-cu.edu.eg,{aboitcairo,aly.fahmy}@gmail.com
[2] Faculty of Computer and Information, Minia University, Egypt
ah.hafez@gmail.com
[3] Scientific Research Group in Egypt (SRGE)
http://www.egyptscience.net

Abstract. Community structure identification in complex networks has been an important research topic in recent years. Community detection can be viewed as an optimization problem in which an objective quality function that captures the intuition of a community as a group of nodes with better internal connectivity than external connectivity is chosen to be optimized. In this paper Artificial Fish Swarm optimization (AFSO) has been used as an effective optimization technique to solve the community detection problem with the advantage that the number of communities is automatically determined in the process. However, the algorithm performance is influenced directly by the quality function used in the optimization process. A comparison is conducted between different popular communities' quality measures and other well-known methods. Experiments on real life networks show the capability of the AFSO to successfully find an optimized community structure based on the quality function used.

Keywords: Networks community detection, Community detection, Social Networks, Fish Swarm optimization.

1 Introduction

A social network is a graph made of nodes that are connected by one or more specific types of relationships, such as values, friendship, work. The goal of community detection in networks is to identify the communities by only using the information embedded in the network topology. Many methods have been developed for the community detection problem. These methods use tools and techniques from disciplines like physics, biology, applied mathematics, and computer and social sciences [1].

One of the special interests in social network analysis is finding community structure. Community is a group of nodes that are tightly connected to each other and loosely connected with other nodes. Community detection is the process of network clustering into similar groups or clusters. Community detection

D. Filev et al. (eds.), *Intelligent Systems'2014*,
Advances in Intelligent Systems and Computing 323, DOI: 10.1007/978-3-319-11310-4_44

has many applications including realization of the network structure, detecting communities of special interest, visualization [2], etc [3].

One of the novel techniques in community detection is Girvan-Newman (GN) algorithm [4]. Girvan-Newman is a divisive technique that uses the edge betweenness as a measure to identify the boundaries of communities. This metric detects the edges between communities by counting the number of shortest paths between two particular nodes that passes through a special edge or node. Later on Girvan and Newman introduced a new technique called Modularity [5]. Modularity measures the quality of a partition of the network, where high Modularity indicates strong community structure that has dense inter-connections between the community nodes, Therefore the community detection problem became a Modularity Maximization problem. Finding the optimal Modularity is an NP-Complete problem, a lot of heuristic search techniques have been investigated to solve this problem such as genetic algorithm (GA), simulated annealing, artificial bee colony optimization (ABC) [1].

The remainder of this paper is organized as follows. In Section 2 we define the community problem and introduce the objective functions used in the research. In Section 3 we describe The Basic AFSO algorithm. In Section 4 we describe our proposed algorithm. Section 5 shows our experimental result on real life social networks. We then offer conclusions in section 6.

2 The Community Detection Problem

A social network can be modeled as a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is a set of nodes, and $\boldsymbol{E}$ is a set of edges that connect two elements of $\boldsymbol{V}$. A community structure $\boldsymbol{S}$ in a network is a set of groups of nodes having a high density of edges among the nodes and a lower density of edges between different groups. The problem of detecting k communities in a network, where the number k is unknown can be formulated as finding a partitioning of the nodes in k subsets that best satisfy a given a quality measure of communities $\mathbf{F}(\boldsymbol{S})$. The problem can be viewed as an optimization problem in which one usually wants to optimize the given quality measure $\mathbf{F}(\boldsymbol{S})$. A single objective optimization problem $(\Omega;\mathbf{F})$ is defined as in the equation 1.

$$Max \;\; \mathbf{F}(\boldsymbol{S}), \; s.t \; \boldsymbol{S} \in \Omega \tag{1}$$

Where $\mathbf{F}(\boldsymbol{S})$ is an objective function that needs to be optimized, and Ω=$\{\mathbf{S}_1, \mathbf{S}_2 \ldots, \mathbf{S}_k\}$ is the set of feasible community structures in a network. A formal definition of the optimization problem is given in [6].

The objective function plays an important role in the optimization process; it's the "steering wheel" in the process that leads to good solutions. Many objective functions have been proposed to capture the intuition of communities, and there is no straight-forward way to compare these objective functions based on their definitions. We use some objective functions that capture this intuition and/or are popular in the literature, and can potentially be used for community detection which are Modularity [5], Community Score [7] and Community Fitness [8]. A detailed description of objective functions can be found in [9] and a similarity comparison can be found in [9–11]. Community Score and Community Fitness both have a positive real-valued parameter that controls the size of the communities.

3 Artificial Fish Swarm Algorithm

Artificial Fish Swarm Algorithm (AFSA), which was presented by X. L. Li [12], is a new swarm intelligence optimization method by simulating fish swarm behavior. It is becoming a prospective method because of its good performances in solving many applications [13].

Applying to the optimization problem, generally a 'fish' represents an individual point in a population. The fish swarm movements seem randomly defined and yet they are objectively synchronized. Fishes desire to stay close to the swarm, to protect themselves from predators and to look for food, and to avoid collisions within the group. Inspired by these behavior, researchers aim to solve optimization problems in an efficient manner. The behavioral model-based optimization algorithms seek to imitate, as well as to make variations on the swarm behavior such as praying, swarming, and following in nature, and to create new types of abstract movements. The environment in which the artificial fish (AF) lives is mainly the solution space and the states of other artificial fish. Its next behavior depends on its current state and its environmental state (including the quality of the question solutions at present and the states of other companions), and it influences the environment via its own activities and other companions activities [13].

The AF realizes external perception by its vision shown in Figure.1. If the state at the visual position is better than the current state, it goes forward a step in this direction, and arrives at new better state; otherwise, continues an inspecting tour in the vision. The greater number of inspecting tour the AF does, the more knowledge about overall states of the vision the AF obtains. Certainly, it does not need to travel throughout complex or infinite states, which is helpful to find the global optimum by allowing certain local optimum with some uncertainty [13].

Let the state vector of artificial fish x consists of n variables such that $x = (x^1, x^2, \ldots, x^n)$, x be the current state of an AF and x_v is new state or a neighbor AF in the visual of x selected according to equation 2 then the basic movement process can be expressed as in equation 3. Where $rand()$ produces random numbers between zero and 1, $Step$ is the step size of a move and $dis(x_i, x_j)$ is a distance measure between two AFs normally it would be the Euclidean distance for a traditional problem.

$$x_v = x + Visual * rand() \tag{2}$$

$$x_{next} = x + \frac{x_v - x}{dis(x_v, x)} * Step * rand() \tag{3}$$

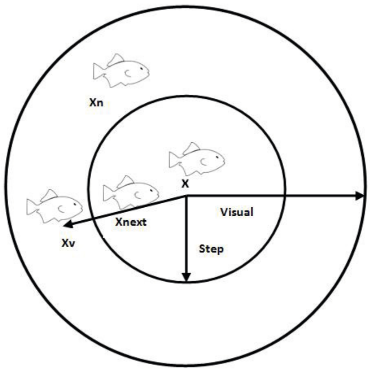


Fig. 1. Shows the Visual and Step of an Artificial Fish

4 Proposed Algorithm

Apply the Basic AFSA directly to the community detection problem is not feasible. The algorithm needs to be redesigned. In this section we describe the modified AFSA so it can be applied to the community detection problem. The algorithm is outlined in listing 1.

4.1 AFSA Parameters Description

The first step of designing an AFSA for solving community detection problem is to devise a suitable representation scheme of an individual AF in a population. The locus-based adjacency encoding scheme [14, 15] is chosen to represent the solution. In this representation, each AF state x consists of n elements $(x^1, x^2, \ldots, x^n)$ and each element can take a value j in the range $[1\ ..\ n]$. A value j assigned to the ith element is interpreted as a link between node i and node j. This means that, in the detected community structure, nodes i and j will be in the same community.

Normally the fish swarm will contain np AFs. The fish current state represents a solution in the search space and the fitness value of the solution represents the amount of food resource at that location. The food concentration in the position of an AF is expressed as $y_i = f(x_i)$, Where y_i is the objective function value associated with x_i. Each AF x_i will have a number of companions np_{fi} which is the number AFs in in x_i's visual satisfying the condition $dis(x_i, x_j) \leq Visual$. The AF x_i' neighborhood field is not crowded and can accommodate more AFs if $np_{fi}/np < \delta$ where δ is the crowd factor which limits the scale of swarms, and more AFs only cluster at the optimal area, which ensures that AF move to optimum in a wide field.

Step: represents the number of modified nodes' membership to move a solution x_i in the direction of a solution x_j as illustrated in Fig.2, where $Step \in [1\ ..\ n]$. The basic movement described in equation 3 can not be applied in our problem. So we use a crossover operator used in Genetic Algorithms [7, 11] where the

Input: A Network $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$
Output: Community membership assignments for network's nodes

```
Initialize the parameters: Visual, Step, Swarm size np, Crowd Factor δ, eps,
the maximum number of iterations Max_Iterations, the number of preying
trials try_number
Randomly initialize each AF in the swarm with a random possible solution as
its current state, and calculate its fitness
repeat
    Memorize the best solution in the swarm
    foreach x_i in the swarm do
        Calculate the distance between x_i and all other AFs
        Calculate np_fi and Set x_i's companions
    end
    foreach x_i in the swarm do
        /* Select the appropriate behavior                    */
        Try Following Behavior
        if No improvement then
            Try Swarming Behavior
            if No improvement then
                Try Preying Behavior
            end
        end
    end
    if —best_Solution(η)-best_Solution(τ)—¡eps then
        Select a random AF from the swarm and execute leaping behavior
    end
    t ← t + 1
until t > Max_Iterations;
return the best solution achieved
```

Algorithm 1. Artificial fish swarm optimization algorithm

mixing ratio is the *Step* size of the move. So in order to move an AF x_i to an AF x_j $move(x_i, x_j)$, the two AFs in the crossover operator are considered as the parents of the new offspring (new AF state). Where the new AF state has a randomly chosen *Steps* optimizing variables from x_j and the rest are from x_i as illustrated in Fig.2.

$Visual$: is a selected constant number $\in [0, 1]$. Since there is no straight way to measure distance between two AFs, we suggested a distance measure $dis(x_i, x_j)$ based on Normalized Mutual Information (NMI) [16] as an indicator on how mush two solutions are closed to each other calculated as in equation 4. NMI is a similarity measure proved to be reliable by Danon et al. [16].

$$dis(x_i, x_j) = 1 - NMI(C(x_i), C(x_j)) \tag{4}$$

where $C(x_i)$ is a functions that decode the AF state back to a community structure and $NMI(c_1, c_2)$ calculate the NMI similarity between two community structures.

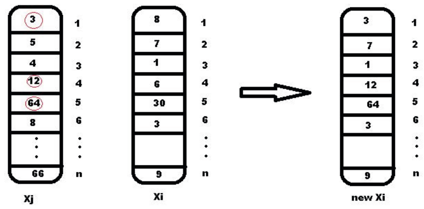


Fig. 2. Shows how the solution X_i moves to the solution X_j with Step of 3

4.2 AFSA Behaviors Description

Preying Behavior: This is a basic behavior that tries to move the food source; generally the fish perceives the concentration of food in water to determine the movement by vision or sense and then chooses the tendency. Let x_i be the AF current state, x_j a randomly select state in x_i's visual where $dis(x_i, x_j) \leq Visual$, and y_i, y_j are the food concentrations (objective function values) related to x_i and x_j respectively, the greater the $Visual$ is, the more easily the AF finds the global extreme value and converges [13]. If $y_i < y_j$ in the maximization problem, it move forward in this direction $move(x_i, x_j)$; Otherwise, select a another state x_j randomly again and judge whether it satisfies the forward condition. If it cannot satisfy after a number of trials try_number, it moves a step randomly. When the try_number is small, the AF can swim randomly, which makes it flee from the local optimum value field.

Swarming Behavior: The fish will assemble in groups naturally in the moving process, which is a kind of living habits in order to guarantee the existence of the colony and avoid dangers. Let x_i be the AF current state, x_c be the center position and np_{fc} be the number of its companions in the current neighborhood where $d_{ij} < Visual$, np is total fish number. If $y_c > y_i$ and $np_{fc}/np < \delta$, which means that the companion center has more food (higher fitness function value) and is not very crowded, then it move to the companion center $move(x_i, x_c)$; Otherwise, executes the preying behavior. Calculating the center position of an AF's companions requires first to calculate the average distance $avgDis_i$ of its companions using equation 5. Then we select the AF with the closest distance value to average distance according to equation 6.

$$avgDis_i = \frac{\sum_j dis(x_i, x_j)}{np_{fi}} \;; such\ that\ dis(x_i, x_j) < Visual \tag{5}$$

$$c = argmin_j \{avgDis_i - dis(x_i, x_j)\}\ \forall j\ subject\ to:\ dis(x_i, x_j) \geq avgDis_i \tag{6}$$

Following Behavior: In the moving process of the fish swarm, when a single fish or several ones find food, the neighborhood partners will trail and reach the food quickly. Behavior description: Let x_i be the AF current state, and it explores the companion x_j in the neighborhood, which has the greatest y_j. If $y_j > y_i$ and $np_{fj}/np < \delta$, which means that the companion x_j state has higher food concentration (higher fitness function value) and the surroundings is not very crowded, then x_i goes forward to the companion x_j $move(x_i, x_j)$; Otherwise, executes the preying behavior.

Leaping Behavior: Fish stop somewhere in water, every AF's behavior result will gradually be the same. If the difference of objective values (food concentration,) become smaller within some iterations, this might mean that the AF is falling into local extreme. So if the objective function is almost the same or difference of the objective functions is smaller than a proportion *eps* during a given $(\eta - \tau)$ iterations, then a randomly selected fish in the whole fish swarm performs a random move.

Since the algorithm employs stochastic process to find optimal solution, it may converge to different solutions (non-deterministic). It is therefore not uncommon to run the algorithm multiple T times i.e. number of restarts, starting with initial different population in each iteration (chosen randomly) and then returning the best solution found across all runs according to the objective function used in the optimization process.

5 Experimental Results

In the section we tested our algorithm on a real life social networks for which a ground truth communities partitions is known. To compare the accuracy of the resulting community structures; we used Normalized Mutual Information (NMI) [16] to measure the similarity between the true community structures and the detected ones. Since Modularity is a popular community quality measure used extensively in community detection, we used it as a quality measure for the result community structure of all other objectives.

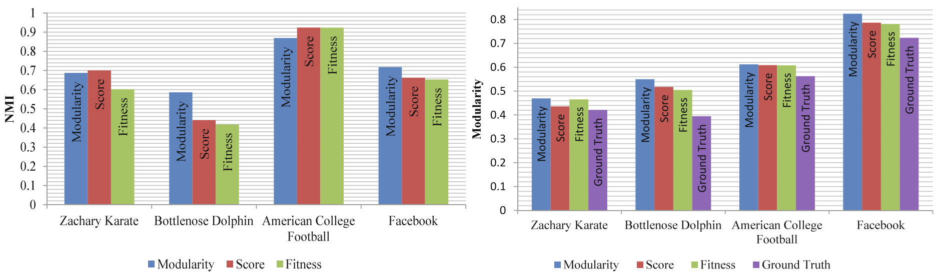
We applied our algorithm on the following social networks datasets :-

- **The Zachary Karate Club**: which was first analyzed in [17], contains the community structure of a karate club. The network consists of 34 nodes. Due to a conflict between the club president and the karate instructor, the network is divided into two approximately equal groups. The network consists of 34 nodes and 78 edges.
- **The Bottlenose Dolphin network**: was compiled by Lusseau [18] and is based on observations over a period of seven years of the behavior of 62 bottlenose dolphins living in Doubtful Sound, New Zealand. The network split naturally into two large groups.
- **American College football network**: [4] represent football games between American colleges during a regular season in Fall 2000, nodes in the graph represent teams and edges represent regular-season games between the two teams they connect. What makes this network interesting is that it incorporates a known community structure. The teams are divided into conferences containing around 8–12 teams each. Games are more frequent between members of the same conference than between members of different conferences, the network is divided into 12 conferences.
- **Facebook Dataset**: Leskovec [19] collects some data for the Facebook website -10 ego networks-. The data was collected from survey participants using

a Facebook application [9]. The ago network consist of a user's –the ego node– friends and their connections to each other. The 10 Facebook ego networks from [19] are combined into one big network. The result network is undirected network which contain 3959 nodes and 84243 edges. Despite there is no clear community structure for the network, a ground truth structure was suggested in [20].

For each dataset; we applied the algorithm with each objective 10 restarts and calculated the NMI and Modularity value of the best solution selected. This process was repeated 10 times and average NMI and average Modularity is reported. The AFSA algorithm was applied with the following parameters values; $Visual = 0.8$, $Step = 0.2 * n$, Swarm size $np = 50$, Crowd Factor $\delta = 0.3$, $eps = 0.001$, the maximum number of iterations $Max\_Iterations = 100$, and the number of preying trials $try\_number = 10$.

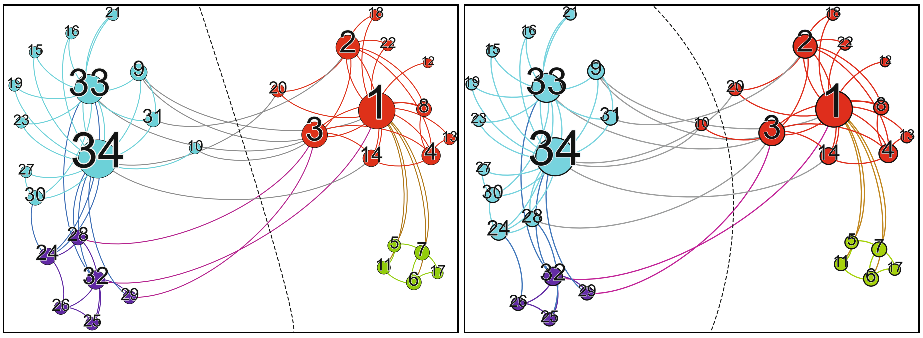
Figure 3a show the average NMI value for each objective when the AFSA is applied with different objectives. Also Fig.3b shows the corresponding average Modularity values for the community structures detected by each objective along with the Modularity value of the ground truth of each network. We can observe that the Modularity objective achieves high NMI values for all social networks. On the other hand; Fitness and Score objectives achieve good NMI values for all social networks except for Bottlenose Dolphin network. The corresponding Modularity value of the community structures detected by Fitness, Modularity and Score objectives are higher than the Modularity value of the ground truth division of those networks as shown in Fig.3b which means in term of Modularity the detected structure is more modular than the original structure.



(a) NMI values **(b)** Modularity values

Fig. 3. Average NMI and Modularity values of the result community structure by each objective on each real social network

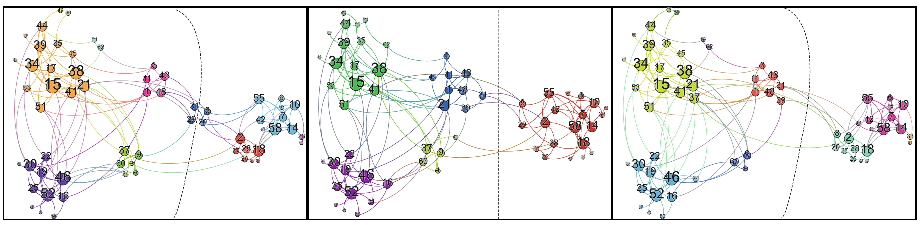
To better understand the behavior of each objective we visualized the detected network divisions produced by each objective on the small size dataset. Figure 4 shows a visualization of the result for the Zachary network. The original division of the network is indicated by the dashed line and the detected structure is indicated by the nodes' colors. All objectives produce a similar result to each other which divide the network into 4 communities with a high Modularity value. The result of Modularity objective is shown in Fig.4a, we can observe in the top level the result is similar to the original division of the network, however in the result structure each group is farther divided into two groups . The result of Fitness is shown in Fig.4b as we can observe in a top level the network is divided into two groups left/right similar the original division of the network however node number 10 is misclassified, farther more each large group is divided into two groups Fig.4b. The result of Score objective is similar to the result of Fitness objective in Fig.4b except of node number 10 is moved to the community on the left.



(a) Modularity **(b)** Fitness

Fig. 4. Visualizations of the result for the Zachary network obtained by different objectives

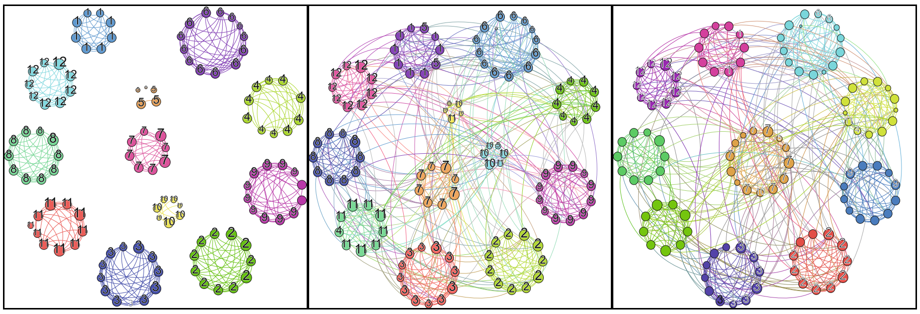
Figure 5 visualizes the result for the Dolphin network. Fitness, Modularity and Score objectives as before are able to detected groups which are more community alike with high Modularity value shown in Fig.5a, 5b and 5c respectively.



(a) Fitness **(b)** Modularity **(c)** Score

Fig. 5. Visualizations of the result for the Dolphin network obtained by different objectives

The result obtained for the College football network by different objectives are visualized in Fig.6. The original division of the network into conferences is highlighted in Fig.6a; only edges between nodes from the same group are shown and nodes' labels refer to which groups they belong to. From Fig.6a we can observe that some nodes were assign to its group however they never played any match with other nodes from their group for example group number 5. Fitness and Score objectives produce a similar result shown in 6b. Modularity objective detected a community structure with 10 communities which assigns nodes from the smaller groups (10,5) into a larger groups leading to a more modular structure Fig.6c.



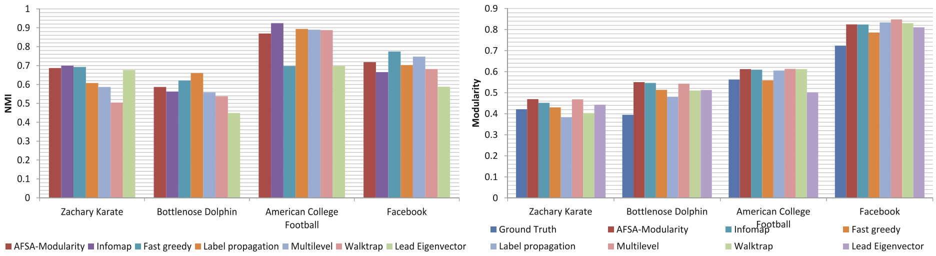
(a) Original Division **(b)** Score **(c)** Modularity

Fig. 6. Visualizations of the result for the American College football network obtained by different objectives

5.1 Comparison Analysis

From the previous section, we noticed that the Modularity objective outperforms Fitness and Score objectives in term of NMI and Modularity. Now we compare the result obtained by AFSA using Modularity objective with other well-known methods in the literature. We selected widely used 6-methods which are Infomap [21], Fast greedy [22], Label propagation [23], Maulilevel or Louvain [24], Walktrap [25] and leading Eignvector [26]. Each method is run 10 times for each dataset and the average NMI and Modularity of the result community structure is reported.

Figure.7a summarize the NMI and Modularity values for all methods. As we can observe that in term of NMI; AFSA produce a good result compared to other methods shown in Fig.7a. In term of Modularity; AFSA is very competitive with other methods shown in Fig.7b. For the small size data set we can observe the AFSA produce a community structure with a high Modularity value compared to all other methods. Regarding the Facebook datasets; AFSA competes with other methods with a very small variance.



(a) NMI Values. **(b)** Modularity Values.

Fig. 7. NMI and Modularity values for each dataset reported by each method

6 Conclusions and Future Work

Artificial Fish Swarm algorithm (AFSA) as an optimization technique works effectively for the community detection problem. However, AFSA's performance is influenced directly by the objective quality function used in the optimization process. Many objective functions have been proposed to capture the intuition of communities which has been used in the literature; hence AFSA was applied with 3 different quality functions as objective functions in order to evaluate their performance which are Community Fitness, Community Score and Modularity. The locus-based adjacency encoding scheme is applied to represent a community structure. The locus-based adjacency encoding scheme has a major advantage that it enables the algorithm to deduce the number of communities k without prior knowledge about it. The results demonstrate that the performance of the proposed approach is promising in terms of accuracy and successfully finds an optimized community structure based on the quality function used. The results show that Modularity objective outperforms the other objectives. A comparison with other popular methods show that AFSA is very competitive with such methods. Visualization of large time varying vector data can utilize importance values in several different ways to, for example, identify unique and abnormal features, enhance visualization as well as enhance both space and time complexity. In the future work, we will focus on studying how to apply our work to visualization problem and address its importance.

References

1. Fortunato, S.: Community detection in graphs. Physics Reports 486, 75–174 (2010)
2. Ali, A.S., Hussien, A.S., Tolba, M.F., Youssef, A.H.: Visualization of large time-varying vector data. In: 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 4, pp. 210–215. IEEE (2010)
3. Masdarolomoor, Z., Azmi, R., Aliakbary, S., Riahi, N.: Finding community structure in complex networks using parallel approach. In: 2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing (EUC), pp. 474–479 (October 2011)

4. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99, 7821–7826 (2002)
5. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physics Rev. E 69, 026113 (2004)
6. Shi, C., Zhong, C., Yan, Z., Cai, Y., Wu, B.: A multi-objective approach for community detection in complex network. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2010)
7. Pizzuti, C.: GA-net: A genetic algorithm for community detection in social networks. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1081–1090. Springer, Heidelberg (2008)
8. Lancichinetti, A., Fortunato, S., Kertesz, J.: Detecting the overlapping and hierarchical community structure of complex networks. arXiv:0805.4770v2 (2008)
9. Leskovec, J., Lang, K.J., Mahoney, M.: Empirical comparison of algorithms for network community detection. In: Proceedings of the 19th International Conference on World Wide Web, pp. 631–640. ACM (2010)
10. Shi, C., Yu, P.S., Cai, Y., Yan, Z., Wu, B.: On selection of objective functions in multi-objective community detection. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 2301–2304. ACM (2011)
11. Hafez, A.I., Al-Shammari, E.T., ella Hassanien, A., Fahmy, A.A.: Genetic algorithms for multi-objective community detection in complex networks. In: Pedrycz, W., Chen, S.-M. (eds.) Social Networks: A Framework of Computational Intelligence. SCI, vol. 526, pp. 145–171. Springer, Heidelberg (2014)
12. Li, X.L., Shao, Z.J., Qian, J.X.: An optimizing method based on autonomous animate: Fish swarm algorithm. System Engineering Theory and Practice 22(11), 32–38 (2002)
13. Neshat, M., Adeli, A., Sepidnam, G., Sargolzaei, M., Toosi, A.N.: A review of artificial fish swarm optimization methods and applications. International Journal on Smart Sensing & Intelligent Systems 5(1), 105 (2012)
14. Shi, C., Zhong, C., Yan, Z., Cai, Y., Wu, B.: A new genetic algorithm for community detection. Complex Sciences 5, 1298–1309 (2009)
15. Pizzuti, C.: Community detection in social networks with genetic algorithms, Atlanta, GA, USA, pp. 1137–1138 (2008)
16. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment 9, 09008 (2005)
17. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)
18. Lusseau, D.: The emergent properties of dolphin social network. Proceedings of the Royal Society of London. Series B: Biological Sciences 270, S186–S188 (2003)
19. McAuley, J.J., Leskovec, J.: Learning to discover social circles in ego networks, pp. 548–556 (2012)
20. Hafez, A.I., Hassanien, A.E., Fahmy, A.A.: Testing community detection algorithms: A closer look at datasets. In: Panda, M., Dehuri, S., Wang, G.-N. (eds.) Social Networking. ISRL, vol. 65, pp. 85–99. Springer, Heidelberg (2014)
21. Rosvall, M., Axelsson, D., Bergstrom, C.T.: The map equation. The European Physical Journal Special Topics 178(1), 13–23 (2009)
22. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical review E 70(6), 066111 (2004)

23. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Physical Review E 76(3), 36106 (2007)
24. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (2008)
25. Pons, P., Latapy, M.: Computing communities in large networks using random walks (long version). ArXiv Physics e-prints, 12 (2005)
26. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E 74(3), 036104 (2006)