Nature inspired algorithms for solving the community detection problem

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

[1] Faculty of Computers and Information, Cairo University, Egypt
eslamhussein.cs@gmail.com ,aboitcairo@gmail.com,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. Nature inspired Swarm algorithms have proven to be effective in solving recent complex optimization problems. Comparing such algorithm is a difficult task due to many facts, the nature of the swarm, the nature of the optimization problem itself and number of controlling parameters of the swarm algorithm. In this work we compared two recent swarm algorithms applied to the community detection problem which are the Bat and Artificial Fish Swarm algorithms. Community detection is an active problem in social network analysis. The problem of detecting communities can be represented as an optimization problem where a quality fitness 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. We also investigated the application of the Bat Algorithm (BA) and Artificial Fish Swarm (AFSA) in solving the community section problem. And introduced a comparative analysis between the two algorithms and other well-known methods. The study show the effectiveness and the limitations of both algorithms.

Keywords: Community detection, Community Structure, Social Networks, Bat algorithm, Artificial Fish Swarm algorithm, Nature-inspired algorithms

1 Introduction

The modern science of networks has brought considerable advances to our understanding of complex systems. One of the most relevant features of graphs representing real systems is community structure, or clustering, i.e. the arrangement of vertices in groups, with plentiful edges linking vertices of the same group and comparatively fewer edges are linking vertices of different groups. Such clusters, groups, or communities, can be seen as independent cluster of a graph. Discovering communities has great importance in biology, sociology, and computer science [1], where systems are usually represented as graphs. Often nodes in the same community share interesting characteristics such as common interest, function or purpose. Therefore community detection is one of the most important problems in network analysis. A 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 has many applications including visualization [2], detecting communities of special interest [3] or image processing [4].

One of the recent techniques in community detection is Girvan-Newman (GN) algorithm [5]. Girvan-Newman is a divisive method that uses the edge betweenness as a measure to discover the boundaries of communities. This measure detects the edges between communities through counting the number of shortest paths between two specific nodes that passes through a special edge or node. Later on Girvan and Newman introduced a new technique called Modularity [6]. Modularity measures the community strength of a partition of the network, where high Modularity means strong community structure that has dense interconnections between the community's nodes, thus the problem of community detection can be viewed as a Modularity Maximization problem. Finding the optimal Modularity is an NP-Complete problem, many heuristic search algorithms have been investigated to solve this problem [1, 7, 8] such as Chicken Swarm Optimization [9], Cuckoo Search Algorithm [10], genetic algorithm (GA) [11, 12] .

The remainder of this paper is organized as follows. In Section 2 we formulate the community detection problem and show the objective function used in the research. In Section 3 we illustrate the basic Artificial Fish Swarm algorithm and basic Bat algorithm. Section 4 shows our experimental result on real life social networks and present the convergence rates and parameters analysis of the algorithms. Finally we present the conclusions in section 5.

2 The community detection problem

A social network can be viewed 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 nodes of $\boldsymbol{V}$. A community structure $\boldsymbol{S}$ in a network is a solution to the problem which is a set of communities of nodes that have a bigger density of edges among the nodes and a smaller density of edges between different sub-groups. the problem of detecting m communities in a network, where the number m is unknown can be formalized as finding a clustering of the nodes in m subsets that can best satisfy a given quality measure of communities $\mathbf{F}(\boldsymbol{S})$. The problem can be formalized as an optimization problem where one usually wants to optimize the given fitness measure $\mathbf{F}(\boldsymbol{S})$ [13].

The objective function has a significant role in the optimization process; it's the "steering wheel" in the process that leads to good solutions. A lot of objective functions have been introduced to capture the intuition of communities, and there does not exist a direct method to compare those objective functions based on their definitions [14–16]. Network Modularity [6] is one of the most used quality measure of communities in the literature. Modularity measures the number of within-community edges relative to a null model of a random graph with the same degree distribution. We use community Modularity as our objective function in the Bat algorithm.

3 Preliminaries

3.1 Artificial Fish Swarm Algorithm

Artificial Fish Swarm Algorithm (AFSA), which was presented by X. L. Li [17], 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 [18], one of them is to solve the community detection problem [19].

The artificial fish (AF) realizes external perception by its vision. 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 [18].

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 1 then the basic movement process can be expressed as in equation 4. 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.

$$x_v = x + Visual * rand() \tag{1}$$

$$x_{next} = x + \frac{x_v - x}{dis(x_v, x)} * Step * rand() \tag{2}$$

The applied AFSA algorithm to the community detection is shown in algorithm1. The algorithm starts with np search agents (AF). Each search agent in the population represent a candidate solution i.e. community structure. The locus-based adjacency encoding scheme [11, 12] is chosen to represent a solution. In this representation, each search agent's 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.

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.

In the AFSA algorithm; *Step* represents the number of modified nodes' membership to move a solution x_i in the direction of a solution x_j, where $Step \in [1 .. n]$. The basic movement described in equation 4 can not be applied in the CD problem. So a crossover operator is used [20, 16] where the mixing ratio is the *Step* size of the move. $Visual$: 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) [21] as an indicator on how mush two solutions are closed to each other calculated as in equation 5.

$$dis(x_i, x_j) = 1 - NMI(C(x_i), C(x_j)) \tag{3}$$

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.

3.2 Bat Algorithm

The Bat Algorithm (BA), that was presented by Xin-She Yang [22], is a new meta-heuristic optimization algorithm derived by simulating the echolocation system of bats. It is becoming a promising method because of its good performances in solving many problems [23], one of them is the community detection problem [24]. In nature, the echolocation behavior of bats used for hunting and navigation, where a bat emits ultrasound pulses to the surrounding environment, then listens back to the echoes in order to locate and identify preys and obstacles. Each bat in the swarm can find the more nutritious area by individual search or moving forward towards a more nutritious location in the swarm. The basic idea of the BA is to imitate the echolocation behavior of bats with local search of bat individual for achieving the global optimum.

At the beginning of the search, each individual bat emits pulses with low frequency but with greater loudness in order to cover bigger area in the search space, later on when a bat approaches from its prey, it increases its pulse emission rate and decreases the pulse loudness, this frequency/loudness adjustment process can control the balance between the exploration and the exploitation operations of the algorithm.

Let the state vector of an artificial bat x composed of n variables such that $x = (x_1, x_2, \ldots, x_n)$, and the velocity vector of artificial bat v composed of n variables such that $v = (v_1, v_2, \ldots, v_n)$, f_i is the current frequency that moves between f_{min} and f_{max}, v_i^t is a new velocity selected according to equation 5 and x_i^t a new state selected as in equation 6.

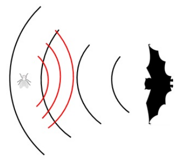


Fig. 1: Shows the echolocation behavior of an Artificial Bat approaching its prey

Where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution, x^* is the current best global solution that is determined after selecting the best solutions between all the n bats in the current population, Because the product $\lambda_i f_i$ is the increase in velocity, then we can either use f_i (or λ_i) to modify the velocity while fixing the other parameter according to the problem type.

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta \tag{4}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*) * f_i \tag{5}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{6}$$

We chose $f_{min} = 0$ and $f_{max} = 1$, and draw initial frequency of each virtual bat from a uniform distribution $f_0 \in [f_{min}, f_{max}]$.

Performing a local search, one of the best solutions in the current population is selected randomly, then a new solution is produced using 7, Where $\epsilon \in [0, 1]$ is random number, and $A_t = \langle A_i^t \rangle$ is the average loudness of all bats in the current population.

$$x_{new} = x_{old} + \epsilon A^t \tag{7}$$

Moreover, the pulse emission rate r and the loudness A of each virtual bat must be updated when a new population generated, As soon as the bat found its prey, its loudness decreases and pulse emission rate increases according to equations 8 and 9

$$A_i^{t+1} = \alpha A^t \tag{8}$$

$$r_i^{t+1} = r_i^0 [1 - exp(-\gamma t)] \tag{9}$$

Where α and γ are constants, after performing some experiments we chose $\alpha = 0.99$, $\gamma = 0.02$, $r_i^0 = 0.01$ and $A_i^0 = 0.99$, Also we can notice that $A_i^t \leftarrow 0$, $r_i^t \leftarrow r_i^0$ as the virtual bat gets closer to its prey $t \leftarrow \infty$, which is rational since a bat has just reached its prey and stops emitting any sound. This update process will only occur when a new better solution generated, which imply that this bat moves towards an optimal solution.

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

The applied BA algorithm to the community detection problem is outlined in listing 2. Similar to AFSA, the BA starts with with np search agents using the same representation. The food condensation in the location of an artificial bat is formulated as $y_i = f(x_i)$, Where y_i is the fitness function value associated with x_i calculated using Modularity quality measure.

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

Initialize the parameters: pulse rates r_i, loudness A_i, Swarm size np, the maximum number of iterations $Max\_Iterations$
Randomly initialize each artificial bat in the swarm with a random possible solution as its current state, the velocity of each bat, and calculate its fitness
repeat
  Generate new solutions by adjusting frequency, and updating velocities and locations/solutions [equations 5 to 7 and the modified equations 10 and 11]
  if $rand > r_i$ **then**
    Select a solution among the best solutions
    Generate a local solution around the selected best solution
  end
  if *rand* $< A_i$ *and* $f(x_i) < f(x_*)$) **then**
    Accept the new solutions
    Increase r_i and decrease A_i
  end
  $t \leftarrow t + 1$
until $t > Max\_Iterations$;
return *the best solution achieved*

Algorithm 2: Bat algorithm

Bat movements are described in equations 4, 5 and 6. In a discrete problem representation such equations can not be applied directly. The difference operator between to bat position will be calculated using equation 10; where $g(x_i)$ is the group assignment of node i in the solution represented by bat position x.

$$d_i = (x_i - x_i^*) = \begin{cases} 1 \; if \; g(x_i) \neq g(x_i^*) \\ -1 \; if \; g(x_i) = g(x_i^*) \end{cases} \tag{10}$$

Now equation 6 will be considered as uniform crossover between (x, x^*) using velocity vector v as the mixing ratio controller, so the new position value will be updated using equation 11.

$$x_i^{new} = \begin{cases} x_i^* \quad if \; v_i \geq 1 \\ x_i \; otherwise \end{cases} \tag{11}$$

In order to create a new solution using equation 7, ϵ should be converted into a vector of size n of uniformly distributed random numbers between 0 and 1, then perform the generation process by selecting a new element from one of the neighbors of the ith node when $\epsilon_i > A^t$, otherwise keep the old neighbor.

The Basic Bat algorithm uses x^* the current global best solution in updating all VB in the swarm, which will lead to moving all bats to the same location. In optimization problems this will might lead to trap the algorithm in a local optima. To overcome this problem instead of using one global best, we select η top bats according to their fitness value, then for each VB update a randomly selected VB from the current η top best is used. η is set 10% of the population size np by trail and error.

4 Experimental results

In the section we tested our algorithm on synthetic network and 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) [21] 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.

For each dataset; we applied the Bat algorithm 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 Bat algorithm was applied with the following parameters values; number of VB in the population $np = 100$ and the maximum number of iterations $Max\_Iterations = 100$.

4.1 Synthetic network

LFR Benchmark is a benchmark proposed by Lancichinetti et al. [25] (LFR benchmark) where the distribution of nodes' degree and communities' size are power laws, with exponents β and γ respectively. Every node shares a fraction $1 - \mu$ of its edges with the other nodes of its community and a fraction μ with the nodes of the other communities; $0 \leq \mu \leq 1$ is the mixing parameter.

Figure 2 show the results testing both AFSA and BA over LFR benchmark while the mixing parameter changes, the networks were generated with the following parameters: Number of nodes = 1000, Exponent for community size distribution = 2, and Exponent for degree sequence = 2.

We can observe in Fig.2, that the modularity is inversely proportional to the mixing parameter. This makes sense, since mixing parameter , is the ratio of the number of external neighbors of each vertex by its total degree. Small values of μ indicate well separated communities, whereas for higher values communities become more and more mixed to each other.

4.2 Real life social networks

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

- **The Zachary Karate Club**: which was first analyzed in [26], 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.

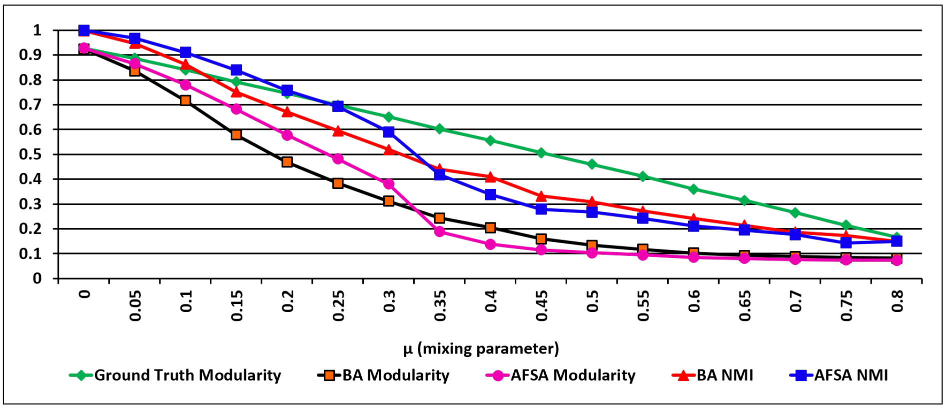


Fig. 2: Different Modularity values and NMI of both BA and AFSA results over different values of mixing parameters.

- **The Bottlenose Dolphin network**: was compiled by Lusseau [27] and is based on observations over a period of seven years of the behaviour of 62 bottlenose dolphins living in Doubtful Sound, New Zealand. The network split naturally into two large groups.
- **American College football network**: [5] 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. 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 [28] collects some data for the Facebook website -10 ego networks-. The data was collected from survey participants using a Facebook application [14]. The ago network consist of a user's –the ego node– friends and their connections to each other. The 10 Facebook ego networks from [28] 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 [29].

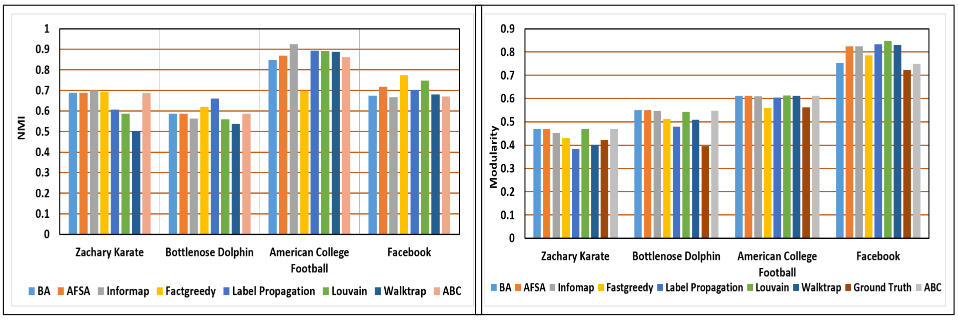
Table 1 summarizes the average NMI and Modularity for the result obtained using the Bat algorithm. We observed that the result for the each network is better that its ground truth in term of Modularity.

Comparative Analysis: Now we compare the result obtained by Bat algorithm with other methods in the literature which are AFSA community detection [19], Infomap [30], Fast greedy [31], Label propagation [32], Multilevel or Louvain [33], Walktrap [34] and leading Eignvector [35]. Each method is run 10 times

Table 1: Modularity Result

	Zachary Karate	Bottlenose Dolphin	College Football	Facebook
Modularity	0.4696	0.5498	0.612	0.753
Ground Truth	0.4213	0.395	0.563	0.7234
NMI	0.6873	0.5867	0.84786	0.67374

for each dataset and the average NMI and Modularity of the result community structure is reported.



(a) NMI values. (b) Modularity values.

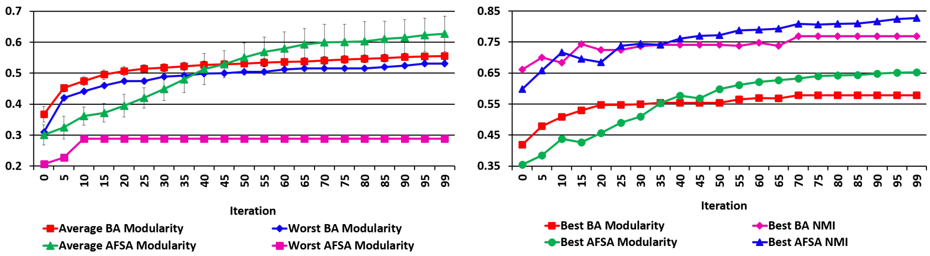
Fig. 3: Average NMI and Modularity values of the result community structure obtained by each algorithm.

Fig.3 summarizes the NMI and Modularity values for all methods. As we can observe that in term of NMI; Bat algorithm produces a good result compared to other methods as shown in Fig.3a. In term of Modularity; Bat algorithm is very competitive with other methods as shown in Fig.3b. For the small size data set we can observe the Bat algorithm produce a community structure with a high Modularity value compared to all other methods. Regarding the Facebook datasets; Bat algorithm failed to produce a result with high Modularity value compared to the other methods.

4.3 Convergence rates

In order to study the convergence rates of the Artificial Fish Swarm Algorithm (AFSA) and the Bat Algorithm (BA), we selected one of the generated LFR networks previously mentioned with the following parameters: Number of nodes = 1000, Exponent for community size distribution = 2, Exponent for degree sequence = 2, and Mixing parameter = 0.25.

In Fig.4, we can observe the convergence rate of the algorithms, the modularity of best/average/worst solutions and NMI of best solution were recorded



(a) Average/worst modularity values. (b) Modularity and NMI values of the best.

Fig. 4: Best/average/worst modularity values and NMI of the best Modularity over successive iterations for BA and AFSA.

every 5 iterations. As we can observe that the modularity of the AFSA starts to converge very slowly after 70% of the iterations. We can deduce that the algorithm continues to explore different locations of the search space after several iterations. This might indicate the ability to escape local optima and navigate in different areas in the search space, reaching to better solutions as a result. As we can observe too, the modularity of the BA converges very slowly and after 25% of the iterations the BA stabilizes and fails to achieve better results. This might indicate the possibility of falling in a local optima. Also we can notice that AFSA outperforms BA although that the initial population/start of BA is much better than of AFSA, which is also proven by the testing results over other real datasets in previous sections.

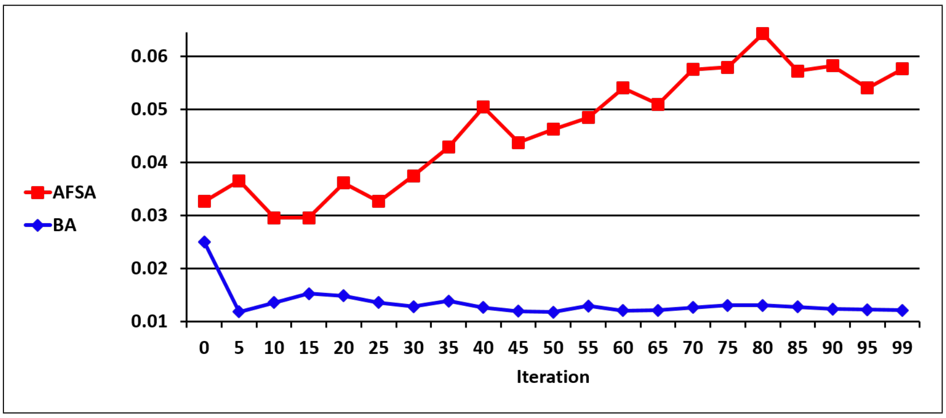
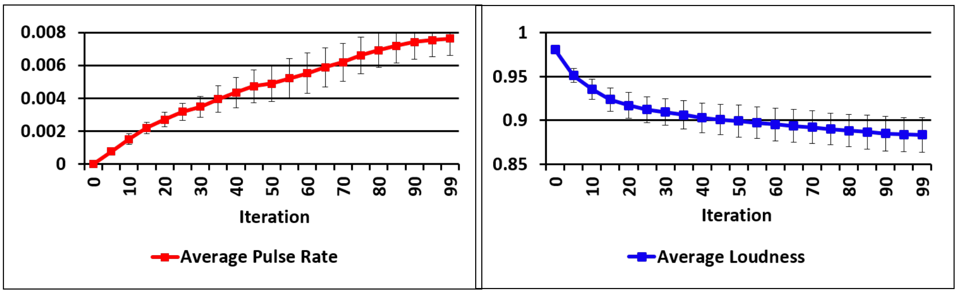


Fig. 5: Standard deviation of the Modularity function over successive iterations (AFSA vs. BA).

In Fig.5; we can observe the values of the standard deviation of the Modularity over successive iterations of both the AFSA and the BA. And we can notice the increase in the standard deviation of AFSA modularity objective function, and that is because AFSA only improves the best virtual fishes in the population and leaves the worst untouched. In Contrary to AFSA, the standard deviation of the BA modularity objective function decreases, and that is because the BA improves most of the population members as we can observe from Fig.4.



(a) Average pulse rate. (b) Average loudness.

Fig. 6: Different values of the average pulse rate and average loudness and their standard deviations over successive populations/iterations.

In Fig.6; We can notice that the increase in pulse rate copes with the decrease in loudness as the normal bat increases its pulse rate the more it becomes close to its prey in order to recognize its properties such as shape/size/exact location, and decreases its loudness since the distance between its current location and its prey location becomes smaller.

4.4 Discussion

As observed from our experimental result that the Bat algorithm performance is promising for small size networks, however for a large networks Bat algorithms performance is degraded compared to other CD algorithms. From our initial analysis, we found there is no much diversity in the VB swarm over the search space and the Bat algorithm does not explore a large region in the search space for the following reasons:

- All the population is moving toward one position (Current global best x^*). over iterations this will lead to all VBs will move/evolve to similar solutions. Despite that we overcome this problem using η top global best, it decreased its impact but did not eliminate it.
- There is no operator/behavior that allow the VB to escape a local optima or jump/explore new random regions in the search space. For example in Genetic algorithm there exist a mutation operation that allow such behavior

even a simple mutation in the current solution could cause a large diversity in the current population. Despite that Bat algorithm performance in other application [23] that are continues in nature is very promising, however for the community detection problem (discrete case) Bat algorithm performance has some limitations.
- The local search and Bat difference operator that we proposed for the community structure are not optimal and it is not clear if they are efficient in exploring the search space. It is possible for another design to cause a significant improvement to the algorithm performance.
- Accepting criteria for new solution has some limitation. The basic Bat algorithm accept new solution only if it is better than the current global best. This may constrain the number of moves that a bat can perform.

5 Conclusions and Future Work

The results of AFSA demonstrate that the performance of the proposed approach is promising in terms of accuracy and successfully finds an optimized community structure. Experimental results demonstrate that the performance of the bat algorithm is quite promising in terms of accuracy and successfully discovers an optimized community structure based on the Modularity quality function for small size networks, yet the performance is getting worse for big size networks. BA algorithm produce good result for the small size network compared to other CD methods, however the result for the larger networks does not compete with other methods.

In future work we are going to conduct an investigation of the discrete BA limitation introduced in Sec.4.4 to propose a new enhanced BA for the community detection problem and investigate other popular bio-inspired optimization algorithms, and apply it for the community detection problem, then conduct an analytical study between those methods to compare their performance. We will also try to expand our work to work on online/dynamic social networks, where the shape and the size of the community changes over time, we will try to investigate the detection of overlapping communities. Online social behavior is not similar to real social networks; because people might connect to those they do not know actually, this behavior might make a community detection algorithm fail to detect a real community structure if it depends only on the connectivity of the network, so we will need to investigate how to solve this problem.

References

1. Santo Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.
2. Ahmed S Ali, Ashraf S Hussien, Mohamed F Tolba, and Ahmed H Youssef. Visualization of large time-varying vector data. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 4, pages 210–215. IEEE, 2010.

3. Michael G Buhnerkempe, Colleen T Webb, Andrew A Merton, John E Buhnerkempe, Geof H Givens, Ryan S Miller, and Jennifer A Hoeting. Identification of migratory bird flyways in north america using community detection on biological networks. *Ecological Applications*, 2016.
4. Youssef Mourchid, Mohammed El Hassouni, and Hocine Cherifi. An image segmentation algorithm based on community detection. In *International Workshop on Complex Networks and their Applications*, pages 821–830. Springer, 2016.
5. M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99:7821–7826, 2002.
6. M.E.J Newman and M. Girvan. Finding and evaluating community structure in networks. *Physics Rev. E*, 69:026113, 2004.
7. Abhay Mahajan and Maninder Kaur. Various approaches of community detection in complex networks: A glance. *International Journal of Information Technology and Computer Science (IJITCS)*, 8(4):35, 2016.
8. Qing Cai, Lijia Ma, Maoguo Gong, and Dayong Tian. A survey on network community detection based on evolutionary computation. *International Journal of Bio-Inspired Computation*, 8(2):84–98, 2016.
9. Khaled Ahmed, Aboul Ella Hassanien, Ehab Ezzat, and Pei-Wei Tsai. An adaptive approach for community detection based on chicken swarm optimization algorithm. In *International Conference on Genetic and Evolutionary Computing*, pages 281–288. Springer, 2016.
10. Ramadan Babers and Aboul Ella Hassanien. A nature-inspired metaheuristic cuckoo search algorithm for community detection in social networks. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, 8(1):50–62, 2017.
11. Chuan Shi, Cha Zhong, Zhenyu Yan, Yanan Cai, and Bin Wu. A new genetic algorithm for community detection. *Complex Sciences*, 5:1298–1309, 2009.
12. Clara Pizzuti. Community detection in social networks with genetic algorithms. pages 1137–1138, Atlanta, GA, USA, 2008.
13. Chuan Shi, Cha Zhong, Zhenyu Yan, Yanan Cai, and Bin Wu. A multi-objective approach for community detection in complex network. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2010.
14. Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.
15. Chuan Shi, Philip S Yu, Yanan Cai, Zhenyu Yan, and Bin Wu. On selection of objective functions in multi-objective community detection. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2301–2304. ACM, 2011.
16. Ahmed Ibrahem Hafez, Eiman Tamah Al-Shammari, Aboul ella Hassanien, and Aly A Fahmy. Genetic algorithms for multi-objective community detection in complex networks. In *Social Networks: A Framework of Computational Intelligence*, pages 145–171. Springer, 2014.
17. X. L. Li, Z. J. Shao, and J. X Qian. An optimizing method based on autonomous animate: Fish swarm algorithm. *System Engineering Theory and Practice*, 22(11):32–38, 2002.
18. Mehdi Neshat, Ali Adeli, Ghodrat Sepidnam, Mehdi Sargolzaei, and Adel Najaran Toosi. A review of artificial fish swarm optimization methods and applications. *International Journal on Smart Sensing & Intelligent Systems*, 5(1):105, 2012.

19. EslamAli Hassan, AhmedIbrahem Hafez, AboulElla Hassanien, and AlyA. Fahmy. Community detection algorithm based on artificial fish swarm optimization. In *Intelligent Systems'2014*, volume 323 of *Advances in Intelligent Systems and Computing*, pages 509–521. Springer International Publishing, 2015.
20. Clara Pizzuti. Ga-net: a genetic algorithm for community detection in social networks, in proc. of the 10th intenational conference on parallel problem solving from nature (ppsn). pages 1081–1090, Dortmund, Germany, 2008.
21. L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 9:09008, 2005.
22. Xin-She Yang. A new metaheuristic bat-inspired algorithm. In JuanR. Gonzlez, DavidAlejandro Pelta, Carlos Cruz, Germn Terrazas, and Natalio Krasnogor, editors, *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, volume 284 of *Studies in Computational Intelligence*, pages 65–74. Springer Berlin Heidelberg, 2010.
23. Xin-She Yang and Xingshi He. Bat algorithm: Literature review and applications. *Int. J. Bio-Inspired Comput.*, 5(3):141–149, July 2013.
24. Eslam A. Hassan, Ahmed Ibrahem Hafez, Aboul Ella Hassanien, and Aly A. Fahmy. *Hybrid Artificial Intelligent Systems: 10th International Conference, HAIS 2015, Bilbao, Spain, June 22-24, 2015, Proceedings*, chapter A Discrete Bat Algorithm for the Community Detection Problem, pages 188–199. Springer International Publishing, Cham, 2015.
25. A. Lancichinetti, S. Fortunato, and J. Kertesz. Detecting the overlapping and hierarchical community structure of complex networks. *arXiv:0805.4770v2*, 2008.
26. W.W Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.
27. D. Lusseau. The emergent properties of dolphin social network. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270:S186–S188, 2003.
28. Julian J. McAuley and Jure Leskovec. Learning to discover social circles in ego networks. pages 548–556, 2012.
29. Ahmed Ibrahem Hafez, Aboul Ella Hassanien, and Aly A Fahmy. Testing community detection algorithms: A closer look at datasets. In *Social Networking*, pages 85–99. Springer, 2014.
30. M. Rosvall, D. Axelsson, and C. T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.
31. Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.
32. Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.
33. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
34. P. Pons and M. Latapy. Computing communities in large networks using random walks (long version). *ArXiv Physics e-prints*, 12 2005.
35. Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

View publication stats