A Discrete Bat Algorithm for the Community Detection Problem

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

Abstract. Community detection in networks has raised an important research topic in recent years. The problem of detecting communities can be modeled as an optimization problem where a quality objective function that captures the intuition of a community as a set of nodes with better internal connectivity than external connectivity is selected to be optimized. In this work the Bat algorithmwas used as an optimization algorithm to solve the community detection problem. Bat algorithm is a new Nature-inspired metaheuristic algorithm that proved its good performance in a variety of applications. However, the algorithm performance is influenced directly by the quality function used in the optimization process. Experiments on real life networks show the ability of the Bat algorithm to successfully discover an optimized community structure based on the quality function used and also demonstrate the limitations of the BA when applied to the community detection problem.

Keywords: Community detection · Community structure · Social networks · Bat algorithm · Nature-inspired algorithms

1 Introduction

A social network can be modeled as a graph composed of nodes that are connected by one or more specific types of relationships, such as friendship, values, work. The purpose of community detection in networks is to recognize 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 different disciplines like physics, chemistry, applied mathematics, and computer and social sciences [1]. One of the main interests in social network analysis is discovering community structure. 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

E. Onieva et al. (Eds.): HAIS 2015, LNAI 9121, pp. 188–199, 2015.
DOI: 10.1007/978-3-319-19644-2_16

applications including visualization, detecting communities of special interest, realization of the network structure [2], etc. [3]. One of the recent techniques in community detection is Girvan-Newman (GN) algorithm [4]. 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 [5]. Modularity measures the community strength of a partition of the network, where high Modularity means strong community structure that has dense inter-connections 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 such as genetic algorithm (GA), simulated annealing, artificial bee colony optimization (ABC) [1]. The remainder of this paper is organized as follows. In Sect. 2 we formulate the community detection problem and show the objective function used in the research. In Sect. 3 we illustrate The Basic Bat algorithm. In Sect. 4 we illustrate our proposed algorithm. Section 5 shows our experimental result on real life social networks. Finally we provide conclusions in Sect. 6.

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 $\mathbf{F}(\boldsymbol{S})$ [6].

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 [7–9]. Network Modularity [5] 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 Bat Algorithm

The Bat Algorithm (BA), that was presented by Xin-She Yang [10], 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 [11].

Biological Inspiration. 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 as shown in Fig. 1. 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.

3.1 Bat Movements Description

Applying the algorithm to the optimization problem, generally a 'bat' represents an individual in a population. The environment in which the artificial bat lives is mainly the solution space and the states of other artificial bats. Its next movement depends on its current state, velocity and its environmental state (including the quality and state of the best bats in the swarm).

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 Eq. 2 and x_i^t a new state selected as in Eq. 3.

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.

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta \tag{1}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*) * f_i \tag{2}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3}$$

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 4, 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.

$$x_{new} = x_{old} + \epsilon A^t \tag{4}$$

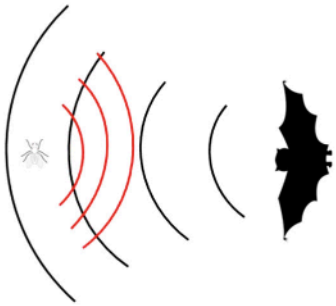


Fig. 1. Shows the echolocation behavior of an Artificial Bat approaching its prey

3.2 Pulse Emission Rate and Loudness Description

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 Eqs. 5 and 6

$$A_i^{t+1} = \alpha A^t \tag{5}$$

$$r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \tag{6}$$

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.

4 Proposed Algorithm

It is not possible to directly apply the Bat algorithm to the problem of community detection. First the algorithm should be redesigned. In the current section we illustrate the modified Bat algorithm so it can be applied to the community detection problem. The algorithm is outlined in listing 1.

4.1 Parameters Redesign

The first step of modifying a BA for solving the problem of community detection, To provide an appropriate scheme of representation of an individual artificial bat in a population. The locus-based adjacency encoding scheme [12,13] is selected to represent the solution. In that representation, each artificial bat state x composed 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 set to the ith element is translated as an association between

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 [Eqs. 2 to 4 and the modified Eqs. 7 and 8]
  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 1. Bat algorithm

node i and node j. Which means that, in the community structure detected, nodes i and j will exist in the same community.

Normally the bat swarm will contain np artificial bats (AB). The bat 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 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.

4.2 Bat Movement

Bat movement are described in Eqs. 1–3. In a discrete problem representation such equations can not be applied directly. First the difference operator between to bat position will be calculated using Eq. 7; where $g(x_i)$ is the group assignment of node i in the solution represented by bat position x.

$$d_i = (x_i - x_i^*) = \begin{cases} 1 \; if \; g(x_i) \neq g(x_i^*) \\ -1 \; if \; g(x_i) = g(x_i^*) \end{cases} \tag{7}$$

Now Eq. 3 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 Eq. 8.

$$x_i^{new} = \begin{cases} x_i^* \;\; if \; v_i \geq 1 \\ x_i \;\; otherwise \end{cases} \tag{8}$$

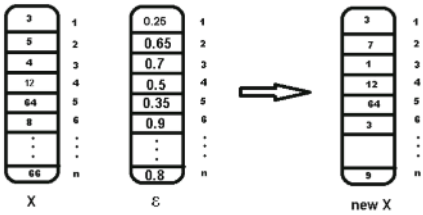


Fig. 2. Shows how a new solution is generated using ϵ and the old solution X when the average loudness $A^t = 0.6$.

4.3 Local Search

In order to create a new solution using Eq. 4, First 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 as illustrated in Fig. 2.

4.4 Current Global Best

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.

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) [14] 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.

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

We applied our algorithm on the following social networks datasets:

- **The Zachary Karate Club:** which was first analyzed in [15], 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 [16] 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:** [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. 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 [17] collects some data for the Facebook website -10 ego networks. The data was collected from survey participants using a Facebook application [7]. The ago network consist of a user's –the ego node– friends and their connections to each other. The 10 Facebook ego networks from [17] 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 [18].

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 $Max\_Iterations = 100$.

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.

The detected community structures for each network is visualized in Fig. 3. Figure 3a shows a visualization of the result for the Zachary network. The original division of the network is indicated by the vertical line and the detected structure

is indicated by the nodes' colors. As we can observe that the detected community structure contains 4 communities with a high Modularity value = 0.4696 in which the top level is similar to the original division of the network, however in the detected structure each group is farther divided into two groups. Thus the NMI of comparing the detected structure with the ground truth of Zachary network is 0.6873 i.e. 68 % of similarity between the two structures. Despite that the NMI value is somehow low, it is not a major judgmental criteria of result for two reasons. First the major quality criteria is Modularity value, since the algorithm is optimizing Modularity objective. Second the know ground truth of the network from the original study [15] might not be the optimal one and it is possible that there is a more modular structure that the ground truth.

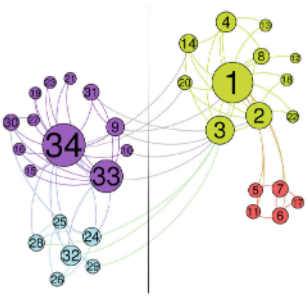
Figure 3b visualizes the result for the Bottlenose Dolphin network. As before the original division of the network is indicated by the curved line and the detected structure is indicated by the nodes' colors. The original division of the Bottlenose Dolphin network is divided into 2 groups with Modularity value of 0.395. The detected community structure by the Bat algorithm is divided into 5 groups and has a higher Modularity value of 0.5498. Regarding the College football network; the original division of the network into conferences is highlighted in Fig. 3c; only edges between nodes from the same group are shown and nodes' color refer to which groups they belong to. From Fig. 3c we can observe that some nodes never played any match with other nodes from their group. The detected community structure for this network is shown in Fig. 3d which contains 10 communities with a more modular structure than the one shown in Fig. 3c.

Figure 3e shows a visualization of the largest 12 communities from the detected community structure of the Facebook dataset. As we can observe that each group show a dense connection between nodes from the same group and spare or low interactions between nodes from different groups except for a few nodes that has a large edge degree cross different groups such as nodes {1750, 476 and 294} which make group membership assignment a hard process for the algorithm.

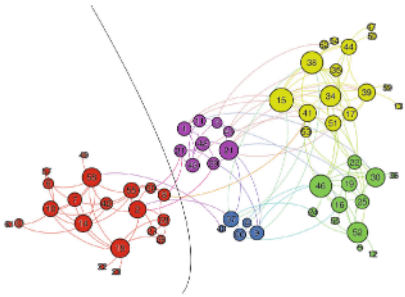
5.1 Comparison Analysis

Now we compare the result obtained by Bat algorithm with other methods in the literature which are AFSA community detecton [19], Infomap [20], Fast greedy [21], Label propagation [22], Multi-level or Louvain [23], Walktrap [24] and leading Eignvector [25]. Each method is run 10 times for each dataset and the average NMI and Modularity of the result community structure is reported.

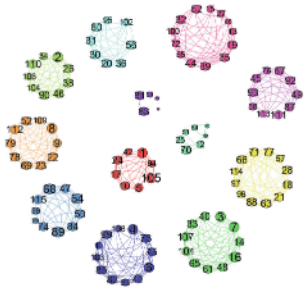
Figure 4 summarize the NMI and Modularity values for all methods. As we can observe that in term of NMI; Bat algorithm produces a good result compared to other methods as shown in Fig. 4a. In term of Modularity; Bat algorithm is very competitive with other methods as shown in Fig. 4b. For the small size 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



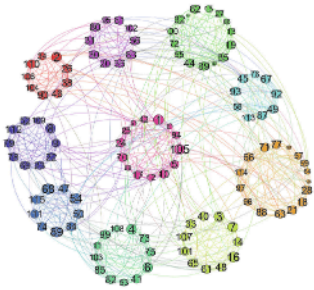
(a) Result for Zachary Karate network.



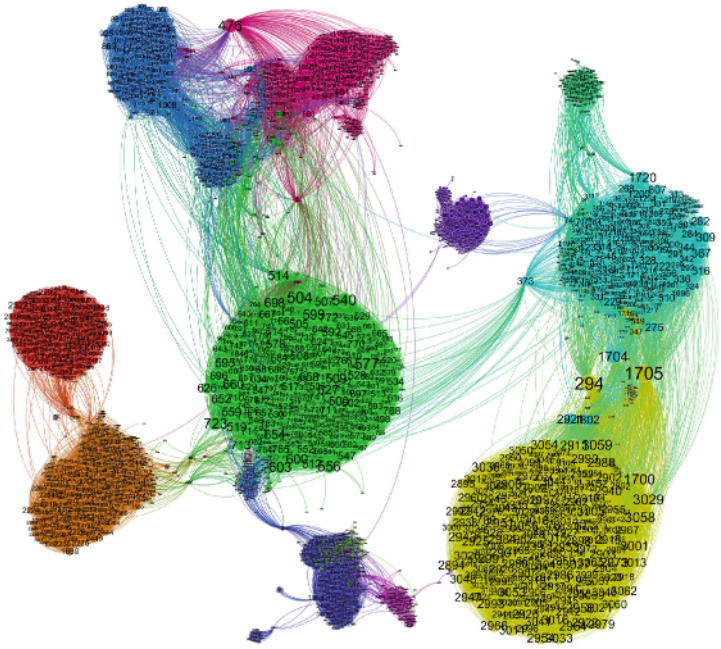
(b) Result for Bottlenose Dolphin network.



(c) Original division of the College Football



(d) Result for the College Football



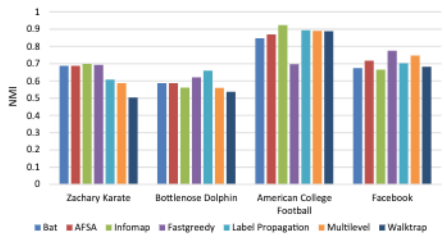
(e) Result for Facebook dataset

Fig. 3. Visualizations of the result obtained by Bat Algorithm on each social network dataset.

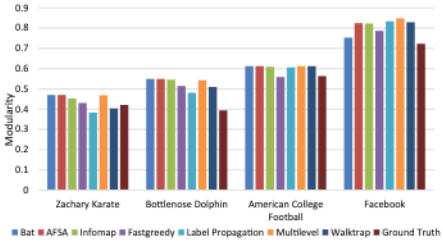
datasets; Bat algorithm failed to produce a result with high Modularity value compared to the other methods.

5.2 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:



(a) NMI values.



(b) Modularity values.

Fig. 4. Average NMI and Modularity values of the result community structure obtained by each algorithm.

- 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 [11] 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 Sects. 4.2 and 4.3 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.

6 Conclusions and Future Work

A discrete Bat algorithm is introduced for finding community structure in social network. 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 past knowledge about it. The BA uses Modularity Quality measure as the fitness function in the optimization process. Experimental results demonstrate that the performance of the bat algorithm is quite promising in terms of accuracy and successfully finds an optimized community structure based on the Modularity quality function for small size networks, however the performance is degraded for large 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 Sect. 5.2 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.

References

1. Fortunato, S.: Community detection in graphs. Phys. Rep. **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. Proc. Nat. Acad. Sci. **99**, 7821–7826 (2002)
5. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. 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. 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)
8. 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)
9. Hafez, A.I., Al-Shammari, E.M., 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. Studies in Computational Intelligence, pp. 145–171. Springer, Heidelberg (2014)

10. Yang, X.-S.: A new metaheuristic bat-inspired algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) NICSO 2010. SCI, vol. 284, pp. 65–74. Springer, Heidelberg (2010)
11. Yang, X.-S., He, X.: Bat algorithm: literature review and applications. Int. J. Bio-Inspired Comput. **5**(3), 141–149 (2013)
12. Shi, C., Wang, Y., Wu, B., Zhong, C.: A new genetic algorithm for community detection. In: Zhou, J. (ed.) Complex 2009. LNICST, vol. 5, pp. 1298–1309. Springer, Heidelberg (2009)
13. 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)
14. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. J. Stat. Mech Theor. Exp. **9**, 9008 (2005)
15. Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**, 452–473 (1977)
16. Lusseau, D.: The emergent properties of dolphin social network. Proc. Roy. Soc. Lond. Ser. B Biol. Sci. **270**, S186–S188 (2003)
17. McAuley, J.J., Leskovec, J.: Learning to discover social circles in ego networks, pp. 548–556 (2012)
18. 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. 87–102. Springer, Heidelberg (2014)
19. Hassan, E.A., Hafez, A.I., Hassanien, A.E., Fahmy, A.A.: Community detection algorithm based on artificial fish swarm optimization. In: Filev, D., et al. (eds.) Intelligent Systems 2014. AISC, pp. 509–521. Springer International Publishing, Heidelberg (2015)
20. Rosvall, M., Axelsson, D., Bergstrom, C.T.: The map equation. Eur. Phy. J. Spec. Top. **178**(1), 13–23 (2009)
21. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)
22. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**(3), 036106 (2007)
23. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theor. Exp. **2008**(10), 10008 (2008)
24. Pons, P., Latapy, M.: Computing communities in large networks using random walks (long version **12** (2005). ArXiv Physics e-prints
25. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**(3), 036104 (2006)