Clustering Social Networks using Nature-inspired BAT Algorithm

Seema Rani[1], Monica Mehrotra[2]
Department of Computer Science, Jamia Millia Islamia, New Delhi, India[1]
Department of Computer Science, Jamia Millia Islamia, New Delhi, India[2]

***Abstract*—The widespread extent of internet availability at low cost impels user activities on social media. As a result, a huge number of networks with a lot of varieties are easily accessible. Community detection is one of the significant tasks to understand the behavior and functionality of such real-world networks. Mathematically, community detection problem has been modeled as an optimization problem and various meta-heuristic approaches have been applied to solve the same. Progressively, many new nature-inspired algorithms have also been explored to handle the diverse optimization problems in the last decade. In this paper, nature-inspired Bat Algorithm (BA) is adopted and a new variant of Discrete Bat algorithm (NVDBA) is recommended to identify the communities from social networks. The recommended scheme does not require the number of communities as a prerequisite. The experiments on a number of real-world networks have been performed to assess the performance of the proposed approach which in turn confirms its validity. The results confirm that the recommended algorithm is competitive with other existing methods and offers promising results for identifying communities in social networks.**

Keywords—Community detection; nature inspired optimization; bat algorithm; discrete particle swarm optimization; social network

I. Introduction

In the era of internet, social network analysis is a lively research area in the field of complex systems. Complex systems in various disciplines can be modeled in the form of networks such as collaboration networks, biological networks, transportation networks and social networks viz. facebook and twitter. Social networks are evolved from communications among users of similar or comparable interests. The basic components of social networks are nodes (vertices) and edges (links) which represent individual users and their associations respectively. Subsequently, social networks are usually represented with the help of graphs. Community detection is one of the fundamental tasks in the course of social network analysis. The target of identifying communities in social networks is interpreted as clustering the group of nodes or divides the network with high cohesive connection strength among the nodes in a group and low among the groups. In other words, the community is a tightly knitted group of nodes and loosely connected with rest of the network. Community detection from social networks supports in understanding their user's topological arrangement and functionality. In fundamental spirit, community detection in social networks has been well-established research domain and has fascinated more researchers from interdisciplinary domains. In modern days, social networks have become part of almost everyone's life. Due to a wide range of available datasets and use in multi-discipline, researchers' interest is increasing in the area of community detection for social network analysis. The community detection problem has been perceived from two different viewpoints, for instance, partitioning/clustering problem and optimization problem.

Girvan and Newman (GN) are founders of recommending the quantitative measure, modularity, for assessing the goodness of community structures in complex networks. Girvan-Newman algorithm [1] is well recognized and widely accepted method in the domain of community detection in complex networks. Due to widespread applicability and usability of community detection in different disciplines for performing diverse tasks, researchers have proposed numerous algorithms with a different line of attack. Some established algorithms include Fast Newman (FN) [2], Clauset, Newman & Moore (CNM)[3], Communities Overlapping based on label PRopagation Algorithm (COPRA) [4], Clique Percolation Method (CPM)[5], Louvain [6] and Label Propagation Algorithm (LPA) [7].

In another view, the community detection problem has been modulated as an optimization problem i.e. modularity maximization problem. Brandes et al. [8] proved that modularity maximization is an NP-hard problem. Meta-heuristic methods are one of preferred choice to deal with NP-hard problems. Many optimization algorithms have been explored to deal with the community detection problem. For example, Genetic Algorithm (GA) [9], Particle Swarm Optimization (PSO) [10] and Ant Colony Optimization (ACO) [11]; optimization techniques exhibit more accurate and promising results.

The growing size of networks and rise in the number of available networks have grabbed the attention of researchers to develop more effective and efficient methods for identifying the community structure in complex networks. For Example, Brentan et al. [8] have used community detection method for proposing District Metered Area (DMA) design in managing large Water Distribution Systems (WDSs). Additionally, they have enhanced it by applying a multilevel particle swarm optimization approach. The proposed approach was tested in a real water supply network and experimental results show evidence for its validity and the significant improvement in obtained results.

Recently, various researchers have applied bat algorithm for finding the solution to community detection problem [12] [13] [14]. Ali et al. [12], presented a discrete bat algorithm for community detection problem. They adopted locus based representation and encoding schema. They redesigned the position update rule for bat movement in respect of community discovery problem. Results show the improved performance in contrast to the ground truth value of networks under test. Chunyu et al. [13], used ordered adjacency list encoding scheme for representation. Their proposed work requires a number of communities as input parameter and experimental results are shown for karate network only. In recent work, Song et al. [14], employed total random initialization scheme for representation. They gave discrete velocity and new position update function. The results show better performance over the existing approaches compared with.

Originally, Bat algorithm was proposed for the continuous optimization problems [15]. Further, Binary bat algorithm was developed for dealing with binary optimization problems [16]. In this work, a new variant named as NVDBA is proposed to deal with community detection problem. In the proposed algorithm, discrete position vector of each virtual bat is coded with node ids' as its element. The associated velocity vector of the bat is encoded as a binary vector i.e. 0 or 1 only will be its members. Discrete bat update rules are redesigned such that position and velocity of each bat get updated with a target that bat moves in the direction of prey/food. To enhance the local search ability of bat, a new solution is generated around the global best solution found so far. Additionally, position vector reshuffle operation [17] is carried out to reduce the number of redundant computations and saves on computational time. Modularity optimization is a remarkable and regularly used approach for community detection in complex networks. Due to its remarkable appearance, the method proposed in the paper considers modularity as the fitness function. Experiments conducted on several real-world datasets show its efficacy and competitiveness against LPA [7], Discrete Particle Swarm Optimization (DPSO) [17] and Discrete Bat Algorithm (DBA) [14] as well. The algorithms are compared in terms of quality of community structure and their statistical significance.

The motivation and major contributions of the proposed work are as follows:

- The work is inspired by the success of new nature-inspired bat algorithm [15] which combines the properties of existing algorithms PSO [18] and simulated annealing [19]. The notion of the proposed work relies on inherent properties of bat algorithm. Bat algorithm performs better than the existing algorithms [14]. The proposed approach is a new variant of Discrete Bat algorithm, referred as NVDBA.
- In this paper, a problem specific, random neighborhood-based initialization scheme is used. Community structure is presented based on network topology and labels of nodes in the network. Bat status update rules are redefined in respect of community detection problem.
- The proposed method does not require a number of communities as a prerequisite. Further, experiments on a variety of datasets validate the performance of the proposed algorithm by comparing it with the traditional method, LPA and evolutionary method, DPSO. In addition, results are also compared with experimental values quoted in recent work [14], referred to DBA.
- The statistical investigation is done by box-plot analysis and Wilcoxon signed-rank test which confirms that proposed algorithm has shown significant improvement.

The organization of this paper is intended as follows. Section 2 presents a background of the related problem, DPSO algorithm, and Bat algorithm. Section 3 provides a detailed description of standard Bat algorithm. In Section 4, the proposed method is described in detail along with its pseudo code and workflow diagram. Further, Section 5 presents workflow of proposed method then followed Section 6 discusses the experimental results obtained on several real-world datasets and its statistical analysis. Finally, Section 7 concludes the paper with a conclusion and future prospects.

II. Related Background

A. *Community Discovery Problem*

A social network is commonly represented by a graph *G* (*V*, *E*) where *V* is the set of nodes and *E* is the set of edges. Each node represents a different user or an entity and each edge represents the relation between the vertices. Community detection problem is defined as partitioning the network into subgraphs such that nodes are connected by a high density of edges in the subgraph and low density of edges among the subgraphs. Mathematically it can be written as; for an input graph *G* = (*V*, *E*) with $V = \{ v_1 , v_2 ,\ldots v_n \}$ and $E = \{e_1, e_2,\ldots\ldots e_m\}$, finding the set of community $C = \{c_1, c_2, \ldots. c_t\}$ where *t* is the number of communities such that $V = c_1 \cup c_2 \cup \ldots \cup c_t$ and $c_1 \cap c_2 \ldots \cap c_t = \phi$. Quality of partitions will be determined by dense intra connection of edges in the community and sparse interconnection of edges from rest of the network. Quantitatively, the most widely used and accepted criteria for measuring the quality of partition is Modularity.

In the undirected graph with no weights and no direction on edges, Modularity *Q* [20] is defined as:

$$Q = \frac{1}{2m}\sum_{i,j}^{n} \left(A_{ij} - \frac{1}{2m}k_i k_j\right) \delta(C_i, C_j) \tag{1}$$

where $k_i = \sum_{j=1}^{n} A_{ij}$ is the degree of ith vertex and. C_i is the community with node i. In case the node i and j belong to the same community, $\delta\left(C_i, C_j\right) = 1$, otherwise the value will be 0. Higher *Q* value indicates the quality of community structure. If all the nodes belong to one community then the value of *Q* = 0. If all the nodes lie in their own community, then the value of *Q* will be negative. The maximum value attainable by *Q* is 1.

The problem of finding t number of communities in the network without a priori input on a number of existing communities with maximum modularity is formulated as modularity optimization problem which is an NP-hard problem. Hence, several evolutionary approaches have been applied and grabbing more attention from a different group of researchers.

B. Discrete Particle Swarm Optimization (DPSO)

Cai et al. [17] have proposed a framework of DPSO for identifying community structures in signed social networks. They identify communities from signed social networks with locus based adjacency initialization strategy. Particle updating rules are defined in following equations:

$$V_i' = Sig\ (\omega V_i + c_1 r_1 (Pbest_i \oplus X_i) + c_2 r_2\ (Gbest\ \oplus X_i)) \quad (2)$$

$$X_i = X_i\ \Theta\ \ V_i{}' \quad (3)$$

$$\omega(k)\ = (\omega_{max}\ -\ \omega_{min})\ \frac{k_{max-}\ k}{k_{max}}\ +\ \omega_{min} \quad (4)$$

In the algorithm, ω_{max} and ω_{min} are taken as 0.9 and 0.4, respectively, and the learning factors c_1 and c_2 are considered as 1.494.

The symbol ''$\oplus$'' in Eq. (2) is the XOR operator and the function *Y*= Sig (*X*) where Y = ($y_1, y_2,$ …., y_n) and X = (x_1, x_2, …., x_n) is defined as

$$y_i = \begin{cases} 1\ if\ rand(0,1) < sigmoid(x_i) \\ 0\ if\ rand(0,1) \geq sigmoid(x_i) \end{cases} \quad (5)$$

And the sigmoid function is given as

$$sigmoid(x) = \frac{1}{1+\ e^{-x}} \quad (6)$$

The operator " Θ " in Eq. (3) provide guidance to the particle so that it will be at a better location in the search space. If X_i = ($x_1, x_2,$ …., x_n) and velocity V_i = ($v_1, v_2,$ …., v_n) then $X' = X_i\ \Theta\ V_i$ will be a new position vector X'=(x_1', x_2',…., x_n') that maps to a new solution such that

$$X_i' = \begin{cases} x_i\ if\ v_i = 0 \\ Nbest_i\ if\ v_i = 1 \end{cases} \quad (7)$$

where $Nbest_i$ is the label identifier possessed by the majority of ith node's neighbors.

$$Nbest_i =\ argmax_r \textstyle\sum_{j \in N} \emptyset(x_{j,}r) \quad (8)$$

In Eq. (8), N = {n_1, n_2,… n_t} is the set of *i^th^* node's neighbours and *r* is an integer which will maximize *f*(*r*).

C. Bat Algorithm

Nature-inspired algorithms have shown the promising result for solving NP-Complete problems. The biological, physical and chemical process of mammals in nature motivates researchers to propose new algorithms and use them for different applications. Out of 1240 species of bats, most of them react and respond to a sophisticated sense of hearing [21]. X. Yang [15] gave a new meta-heuristic algorithm named as Bat algorithm (BA) inspired by echolocation behavior of bats. Mirjalili et al. [16] introduced binary bat algorithm for solving discrete optimization problems. The success of bat algorithm has inspired researchers to explore it to solve different optimization problems. For example, bat algorithm is applied to solve the optimal power flow problem (OPF) [22]. Experimental results obtained with their proposed approach reveal effective and robust high-quality response to the OPF problem. Following, Ssection 3 gives the detailed description of standard bat algorithm.

III. Description of Bat Algorithm

Bat algorithm is proposed by simulating the echolocation characteristics of bats. During simulation of original bat algorithm [15], virtual bats are used and the update rules for each bat are governed as:

1) Update rules: The position vector and velocity vector of each virtual bat is represented as X = (x_1, x_2, …., x_n) and V = (v_1, v_2, …., v_n) composed of n variables. The position vector X represents an instance of the solution for the optimization problem. The current frequency of the bat is denoted by ν_i which ranges between ν_{min} and ν_{max}. The new population will be generated by the movement of bats in search space governed by update rules of their velocity vector V and position vector X given by Eq. (10) and Eq. (11) respectively.

Here, x_{gbest} denotes current global best solution obtained so far from all bats

$$\nu_i = \nu_{min} + (\nu_{max} - \nu_{min})\ \times \beta \quad (9)$$

$$V_i^t = V_i^{t-1} + \left(x_i^{t-1} - x_{gbest}\right)\ \times \nu_i \quad (10)$$

$$x_i^t = x_i^{t-1} +\ V_i^t \quad (11)$$

where $\beta \in [0,1]$ is a uniformly distributed random number in the range of 0 and 1, $\nu_{min} = 0$ and $\nu_{max} = 1$ is chosen.

2) Local Search ability: To boost the local search capability, the new solution will be generated locally using random walk process expressed in Eq. (12).

$$x_{new} = x_{gbest} + \epsilon\ A^t \quad (12)$$

where $\epsilon \in [0,1]$ and A^t = average loudness of all bats in the current population.

3) Update Loudness (A): It will be updated for each virtual bat as iteration proceeds and the new population is generated. When the bat reaches near to their prey/food, its loudness decreases by the following expression:

$$A_i^{t+1} = \propto A^t \quad (13)$$

where $\propto$ is a constant in the range $0 < \propto < 1$ such that $A_i^t \to 0$ as $t \to \infty$. The rate of decrement in loudness is determined by $\propto$

4) Pulse emission rate update(r): In order to search for the best solution, bat explores the search space by updating pulse emission rate. The moment when bat reaches near to its prey, pulse emission rate increases by the following equation:

$$r_i^{t+1} =\ r_i^0(1 - e^{-\gamma t}) \quad (14)$$

where $\gamma > 0$ is a constant such that $r_i^{t+1} \to\ r_i^0$ as $t \to \infty$ and r_i^0 = initial pulse emission rate.

Both, loudness and pulse emission rate gets updated only when a new solution is improved over the previous solution that infers that bats are on the way of a near-optimal solution.

IV. Framework of Proposed Method

In a theoretical sense, community detection is modularity maximization and NP-hard problem. The superiority of bat algorithm inspires and motivates to deal with the modularity maximization problem. The representation schema and bat movement update rules are defined in following subsections.

A. Discrete Bat Position Encoding / Decoding Scheme

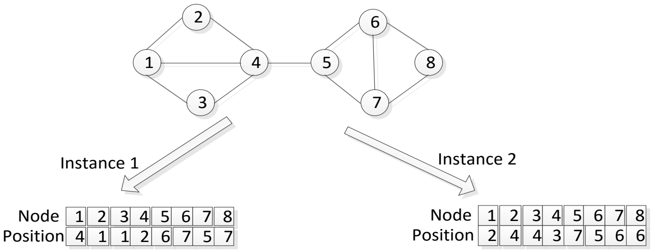
In population-based approaches, population initialization should be in the context of problem domain instead of purely total randomization. In proposed NVDBA, random neighborhood-based encoding is used to represent the position vector of each virtual bat. The chosen encoding scheme for position vector supports in finding the number of communities without using its previous knowledge or a priori input on a number of communities. The position vector of virtual bat represents an instance of the solution to the optimization problem. Discrete position vector is an n-dimensional vector for each virtual bat (say b^{th} bat) in context of community detection problem which is defined as $X_b = \{x_1, x_2, ...,x_n\}$ where $x_i \in [1,n]$ that is set of real integers and n is a total number of nodes in the network. Each dimension of a position vector is a random neighbor of the respective node as depicted in Fig. 1.

If x_i=x_j, then it implies that node i and j belong to the same community as shown in Fig. 2.

B. Discrete Bat Velocity Encoding

The velocity of each virtual bat will be n-dimensional vector written as $V_b = \{v_1,v_2, ...,v_n\}$ where each v_i is binary coded such that $v_i \in \{0,1\}$ and $1\leq i \leq n$ where n denotes the number of nodes in the network. The change in position vector dimension will be governed by a component of velocity vector such that if v_i= 1 then change associated element in position vector else it will remain in the same state.

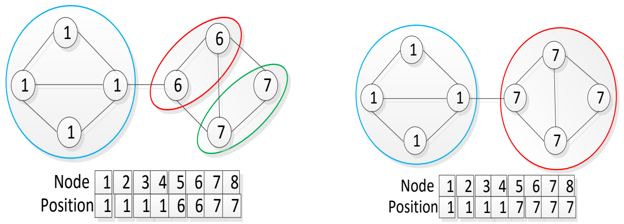
During initialization, it is assumed that dimensions of discrete velocity vector for each virtual bat will be 0 as depicted in Fig. 3.



Node	1	2	3	4	5	6	7	8
Position	4	1	1	2	6	7	5	7

Node	1	2	3	4	5	6	7	8
Position	2	4	4	3	7	5	6	6

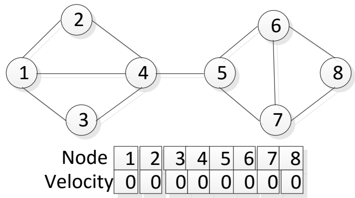
Fig. 1. Position Vector Initialization Instance.



Node	1	2	3	4	5	6	7	8
Position	1	1	1	1	6	6	7	7

Node	1	2	3	4	5	6	7	8
Position	1	1	1	1	7	7	7	7

Fig. 2. Instance of Community Structure.



Node	1	2	3	4	5	6	7	8
Velocity	0	0	0	0	0	0	0	0

Fig. 3. Instance of Velocity Vector Initialization.

C. Illustration of Discrete Bat Status Update Rules

1) Frequency: The frequency in the proposed method is presented by a real number given by Eq. (8) as in original bat algorithm. Initial v_{min} and v_{max} depends on the domain where it is been used.

2) Velocity: The velocity of each bat is represented by an n-dimensional binary vector. A velocity vector component tells the number of bat position that will change at a certain point of time which guides the movement of the bat. If $v_i = 1$ then corresponding element in position vector will be changed else it will remain in the same state. Here, V_i^{t-1} is previous state velocity, x_i^{t-1} is previous state position of bat and x_{gbest} is a current global best solution obtained so far from all bats whereas $x_i^{t-1} \oplus x_{gbest}$ is a comparison between the previous positions with global best position amongst all bats. The bats communicate among themselves via global best position obtained so far and tend to move in the direction of the global best position.

The velocity vector update rule is given in Eq. (15)

$$V_i^t = \tau(V_i^{t-1} + \left(x_i^{t-1} \oplus x_{gbest}\right) \times \boldsymbol{v}_i) \tag{15}$$

In Eq. (15), $\oplus$ is defined as XOR operator and the sigmoid function $\tau(V_i^t)$ maps the real value of velocity to either 0 or 1 defined as below:

$$\tau(V_i^t) = \frac{1}{(1+e^{-V_i^t})} \tag{16}$$

$$\begin{cases} V_i^t = 1 \; if \; U(0,1) < \tau(V_i^t) \\ V_i^t = 0 \; if \; U(0,1) \geq \tau(V_i^t) \end{cases} \tag{17}$$

Here, $U(0,1)$ is uniformly generated random number between 0 and 1.

3) Position: According to above redefined discrete velocity updating strategy, new position update rule is given in Eq. (18)

$$x_i^t = x_i^{t-1} \circledast V_i^t \tag{18}$$

Here, operator $\circledast$ is applied between the previous position and newly defined velocity vector that results to a position vector. The definition of the operator is based on label propagation updating strategy. Given a position vector X = $\{x_1, x_2, ..., x_n\}$ and a velocity vector V = $\{v_1, v_2, ..., v_n\}$ then the new position vector X$_{new}$ = $\{x'_1, x'_2,...., x'_n\}$ is defined as

$$\begin{cases} x'_i = x_i \; if \; v_i = 0 \\ x'_i = C(i) \;\; otherwise \end{cases} \tag{19}$$

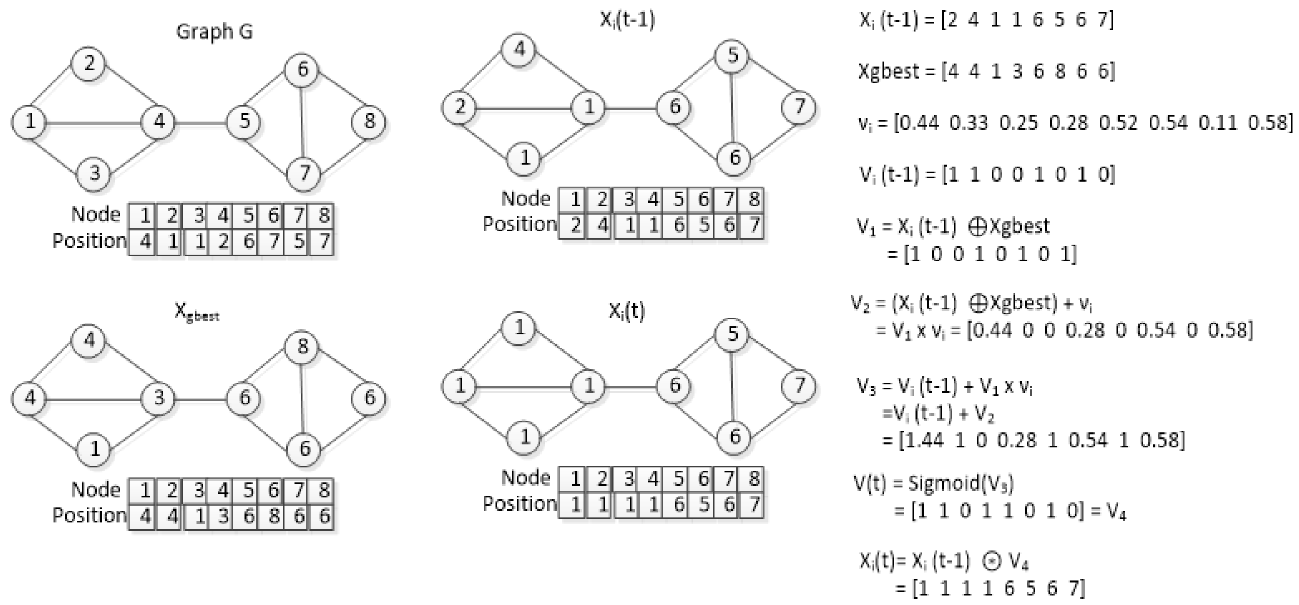


Fig. 4. Bat Status Update Rules.

where, $C(i)$ is the label identifier determined by

$$C(i) = arg\ max_r \sum_{j \in N\ (i)} \psi(x_j, r) \tag{20}$$

Here, $N\ (i)$ is the set of ith node's neighbor. The function $\psi(r)$ returns an integer r that maximizes $\psi(r)$. If more than one value of r maximizes $\psi(r)$ then randomly one among them will be chosen. So, $C(i)$ is an integer that is community identifier possessed by a maximum neighbour of ith node.

An example instance on toy network with visual representation for the definition of discrete velocity and discrete position update rules is illustrated in Fig. 4.

4) Position reshuffle: A position vector at any stage represents an instance of the solution to the optimization problem. At a certain point in time, two different position vectors may correspond to identical community structure that is they have equivalent community structure. For example, an instance X_1={3,3,3,3,8,8,8,8} and X_2={1,1,1,1,7,7,7,7}, they both correspond to same community structure as represented in Fig. 5.

Position restructuring is carried out to reduce redundant computation and save on computational time [17].

D. Local Search

To enhance the local search capability of the bat, a new solution x_{new} is generated from the global best solution x_{gbest} by using Eq. (12). The Eq. (12) cannot be applied directly in the context of discrete position vector for creating the new solution [12]. In the proposed method, a new solution is created based on definition written in Eq. (21)

$$\begin{cases} x_{i\ (new)} = \hat{x_i}\ if\ \epsilon > A^t \\ x_{i\ (new)} = x_{(gbest)i}\ otheriwse \end{cases} \tag{21}$$

Here, $\hat{x_i} \in N(i)$ that is $\hat{x_i}$ is one from a set of ith node's neighbors.

The instance on toy network with a graphical representation for forming a new solution is depicted in Fig. 6.

E. Loudness(A)

The loudness of each virtual bat will be updated by Eq. (13). After performing experiments, the initial value for A and α is initialized with A= 0.95 and $\alpha = 0.95$. As bats reach near to their food, the loudness usually decreases. The rate of decrease in loudness is controlled by the parameter α as shown in Fig. 7.

F. Pulse Emission Rate (r)

The pulse emission rate of each virtual bat is updated by Eq. (14) that tends to find the better solution near a global best solution X_{gbest}. Generally, pulse emission rate increase as the bats reaches near to their pray/food. The rate of increase in pulse emission rate is controlled by the initial value of r_0 and γ. The initial value of r_0 = 0.5 and $\gamma = 0.03$ is assumed and an increase in *r* is depicted in Fig. 8.

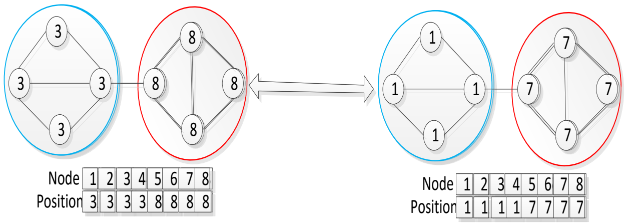
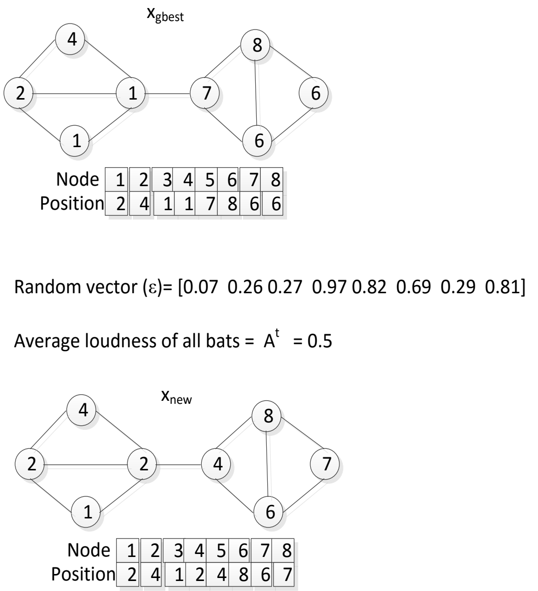


Fig. 5. Graphical Illustration of Position Reshuffles.



Node	1	2	3	4	5	6	7	8
Position	2	4	1	1	7	8	6	6

Random vector (ε)= [0.07 0.26 0.27 0.97 0.82 0.69 0.29 0.81]

Average loudness of all bats = A^t = 0.5

Node	1	2	3	4	5	6	7	8
Position	2	4	1	2	4	8	6	7

Fig. 6. Schematic Illustration of New Solution Generation.

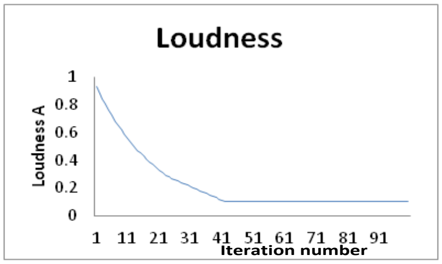


Fig. 7. Change in Loudness with Iteration Number.

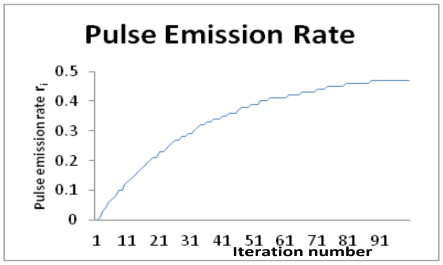


Fig. 8. Change in Pulse Emission Rate with Iteration Number.

V. WORKFLOW OF NVDBA

In this work, discrete position vector and binary velocity vector is adopted. Initialization and updating rule strategies are explained with the help of a graphical example in the preceding subsections. Modularity is one of the prominent representative optimization functions to evaluate the goodness of community partition. Thus, proposed method employs modularity as the fitness function and a number of iterations as the termination criteria. The following Fig. 9 presents the workflow of NVDBA to deal with the community discovery problem.

The proposed NVDBA pseudocode is presented in Table I.

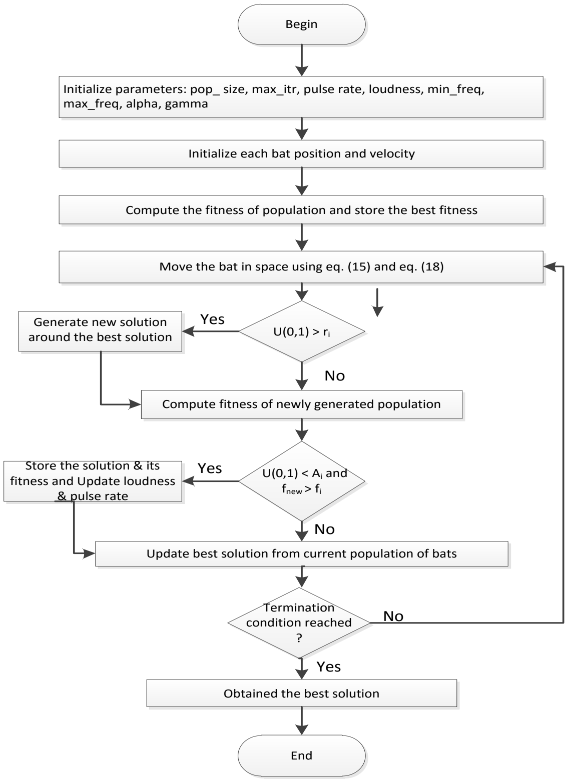

Begin
Initialize parameters: pop_ size, max_itr, pulse rate, loudness, min_freq, max_freq, alpha, gamma
Initialize each bat position and velocity
Compute the fitness of population and store the best fitness
Move the bat in space using eq. (15) and eq. (18)
Generate new solution around the best solution
Yes
U(0,1) > r_i
No
Compute fitness of newly generated population
Store the solution & its fitness and Update loudness & pulse rate
Yes
U(0,1) < A_i and $f_{new} > f_i$
No
Update best solution from current population of bats
Termination condition reached ?
No
Yes
Obtained the best solution
End

Fig. 9. Work Flow Diagram of NVDBA.

TABLE I. PSEUDO-CODE FOR NVDBA

Pseudo-code NVDBA

```
Input: Social Network graph G (V, E)
Output: Best solution &maximum value$f_{max}$=max $f(x)$
initialize the parameter population size = np, maximum number of iteration N, r, A,α, γ , $v_{min}$ and $v_{max}$
initialize bat population
$f$ =evaluate the fitness of population based on objective function (modularity)
$x_{gbest}$= best solution from all bats
while termination condition not met or  $t < N$  do
    for i=1 to np (number of bats) do
        $v_i = v_{min} + (v_{max} - v_{min}) \times \beta$
        $V_i^t = \tau(V_i^{t-1} + (x_i^{t-1} \oplus x_{gbest}) \times v_i)$
        z=$x_i^t = x_i^{t-1} \circledast V_i^t$
        if $U\ (0,1)\ > r_i$ then
            if $\epsilon > A^t$ then
                z= $\hat{x_i}$
            else
                z=$x_{(gbest)i}$
            end if
        end if
        reshuffle(z)
        $f_{new}$ = evaluate fitness (z)
        if $f_{new} > f_i$ and $U\ (0,1) < A_i$ then
            $x_i = z$
            $f_i = f_{new}$
            Increase pulse emission rate (r) by Eq. (14)
            Decrease loudness (A) by Eq. (13)
        end if
    end for
    $x_{gbest}$= update best solution till now from current population of all bats
end while
```

VI. Experimental Observations

In this section, experiments were performed on some standard real networks on a desktop machine with Intel Core i7-3770 CPU @ 3.40 GHz processor and 4 GB of RAM. The operating system is windows 7 OS. The algorithms codes are written in python with networkx and matplotlib. In this paper, DPSO [17] is simulated with random neighborhood-based initialization schema for undirected and unweighted social networks. However, LPA results are obtained by using the inbuilt function of 'igraph' package defined in R. In [14], testing of DBA is performed on three real-world networks and their results are taken for comparison purpose and confirming the improved performance of NVDBA.

Each social network dataset during experimentation is presumed as unweighted and undirected social network. Social network datasets used during experimentation and their basic properties are listed in Table II.

In population-based approach, DPSO parameters are taken similar to [17] for simplicity. In proposed NVDBA, parameters population size and maximum iterations are taken same as in DPSO. Min frequency and Max frequency is initialized with 0 and 1 respectively [12]. Other parameters; initial pulse rate, initial loudness, gamma, and alpha are initialized after prior experimentation in a way that pulse emission rate and loudness constraints hold. The required experimental parameters for DPSO and NVDBA are listed in Table III.

TABLE II. Datasets used in Experimentation

Network name	# nodes	# edges	Ref.
Karate	34	78	[23]
Dolphin	62	159	[24]
Usa	49	107	[25]
Zebra	27	111	[26]
Polbooks	105	441	[27]
Football	115	613	[1]
AIDSblog	146	187	[28]
Jazz	198	2742	[29]
Netscience	1589	2742	[30]

TABLE III. Initial Parameters Value of DPSO and NVDBA

Parameter name	Value	Parameter name	Value
Pop size	100	Pop size	100
Max-iteration	100	Max-iteration	100
w_{max}	0.9	v_{max}	1
w_{min}	0.4	v_{min}	0
C_1	1.494	Initial pulse rate r0	0.5
C_2	1.494	Gamma γ	0.03
		Initial loudness A	0.95
		Alpha α	0.95

With respect to the massive amount of work carried out in the existing literature based on modularity to assess the quality of community partition, the proposed method has adopted it as an evaluation metric. Each algorithm is executed independently 15 times on all dataset considered under test with an idea of reducing the statistical error. For statistical analysis and evaluating the performance; maximum modularity, average modularity, and the standard deviation is reported. The number of communities identified by each algorithm on all the networks under study is in accordance with the maximum value of modularity.

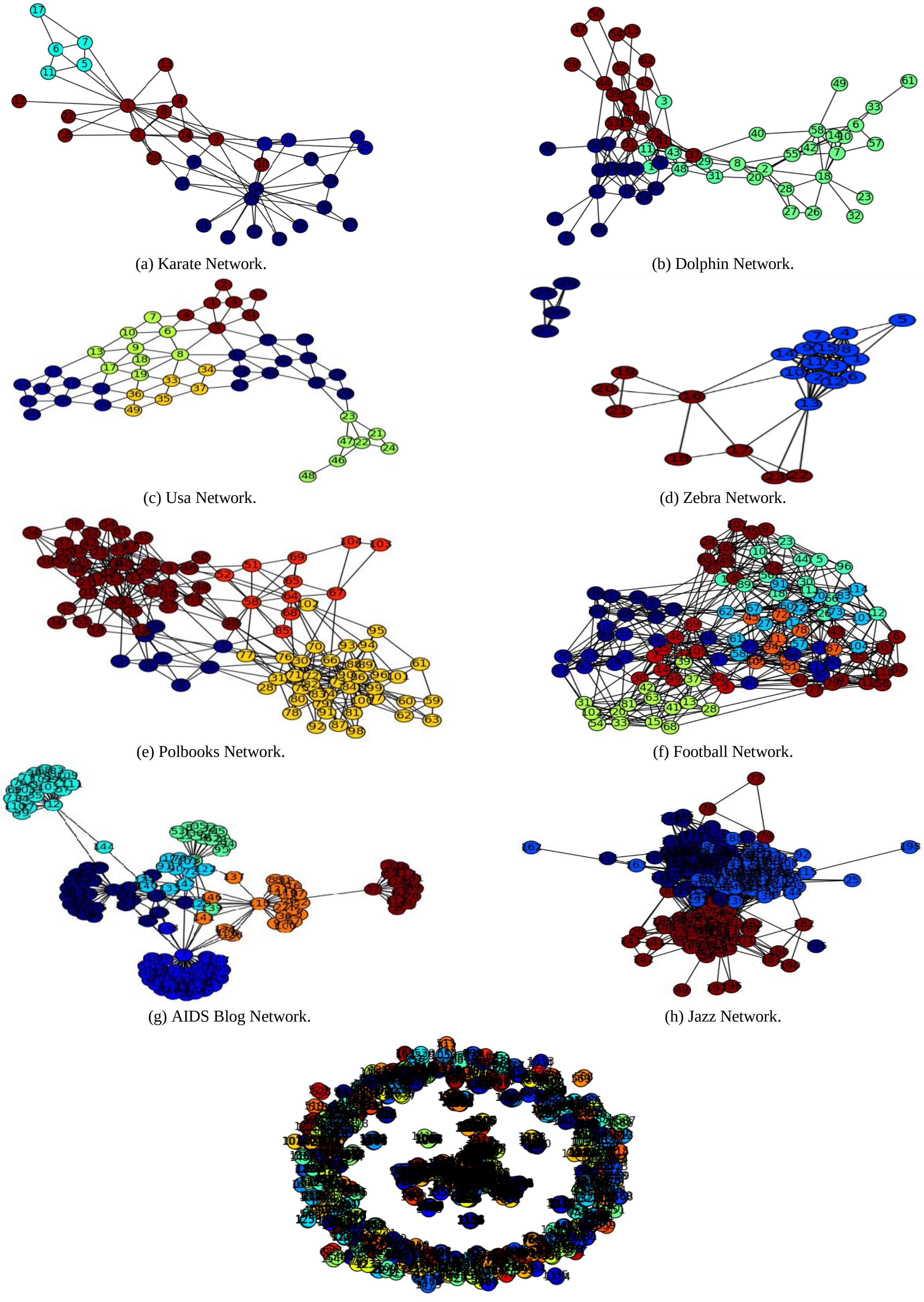
A. Performance Evaluation

The quantitative results based on experimentation performed on real-world networks are reported in Table IV.

The comparative analysis of obtained results reveals the effectiveness of the proposed algorithm in contrast with traditional community detection algorithm (LPA) and nature inspired approach (DPSO). Additionally, it is compared with recent algorithm DBA. Further, observation inferences that NVDBA results in higher objective function values i.e modularity. High modularity characterizes the dense connection in a community and sparse connection among the communities. The community structure obtained by proposed algorithm on each network at highest modularity is shown in Fig. 10.

TABLE IV. Experimental Values on Real Networks

Network Name	Name of Algorithm	# of communities	Max	Avg	Standard Deviation
Karate	LPA DPSO DBA NVDBA	4 3 2 4	0.40 0.3991 0.392 0.4156	0.368 0.3781 0.362 0.4156	0.023 0.0208 -- 0.0000
Dolphin	LPA DPSO NVDBA	4 5 4	0.52 0.5237 0.5265	0.486 0.5089 0.5262	0.036 0.0149 0.0003
Usa	LPA DPSO NVDBA	6 5 6	0.59 0.5855 0.5958	0.534 0.5479 0.5911	0.055 0.0249 0.0037
Zebra	LPA DPSO NVDBA	3 3 3	0.27 0.2702 0.2702	0.257 0.2636 0.2702	0.015 0.0048 0.0000
Polbooks	LPA DPSO DBA NVDBA	3 4 3 4	0.5 0.5148 0.479 0.5262	0.4860 0.5046 0.463 0.5262	0.0168 0.0146 --- 0.0000
Football	LPA DPSO DBA NVDBA	10 11 12 10	0.60 0.6032 0.596 0.6046	0.583 0.5959 0.531 0.6044	0.014 0.0080 -- 0.0001
AIDSblog	LPA DPSO NVDBA	7 7 7	0.64 0.6433 0.6469	0.627 0.6244 0.6438	0.023 0.0116 0.0014
Jazz	LPA DPSO NVDBA	4 3 3	0.44 0.4427 0.4428	0.377 0.3644 0.4428	0.078 0.0799 0.0000
Netscience	LPA DPSO NVDBA	452 303 333	0.92 0.9052 0.9079	0.911 0.8953 0.9047	0.003 0.0056 0.0017



(a) Karate Network.

(b) Dolphin Network.

(c) Usa Network.

(d) Zebra Network.

(e) Polbooks Network.

(f) Football Network.

(g) AIDS Blog Network.

(h) Jazz Network.

(i) Net Science Network.

Fig. 10. Community Structure on Real-World Datasets with the Highest Modularity (Q).

Results recorded in Table IV, demonstrate the rationality of the proposed method. Increase in modularity value for almost all data sets considered during experimentation shows that proposed method enhances the quality of community structures. Also, it points out its statistical significance by computing the average modularity value for 15 independent runs and standard deviation on each dataset. Minimum standard deviation for each data set conveys that the proposed method is reliable. Overall, experimental results demonstrate that NVDBA provides more promising results and is effective as well.

B. Statistical Analysis

In the following, SPSS statistics software is used for performing the statistical analysis. For each network under study, the proposed algorithm is executed 15 times independently. The necessary experimental parameters required for the implementation are listed in Table II. The experimental values are recorded in Table III. The box-plot analysis (Fig. 11) of proposed algorithm portrays variability in experimental values over 15 runs around the mean.

The variation in modularity value obtained during independent runs is realistically low. This box plot analysis concludes that reliability of proposed algorithm is convincingly high.

The proposed algorithm was prospected to improve the quality of community structure in contrast to the existing algorithm. So, for further validating the performance of the proposed algorithm, non-parametric test i.e. Wilcoxon Sign-Rank Test [31] is conducted. DPSO and NVDBA detect community structure from undirected and unweighted real-world networks with the intent of finding the highest modularity. Both algorithms are executed 15 times independently on each dataset with same population initialization methodology. Modularity values for each run on each dataset by both the algorithms are recorded up to four decimal points. Following are the steps performed to response the query: “Change in modularity value by DPSO and NVDBA is not statistically significant?” For answering this query, Wilcoxon Sign-Rank Test is performed on datasets listed in Table II. Firstly, the hypothesis is framed as:independently on each dataset with same population initialization methodology. Modularity values for each run on each dataset by both the algorithms are recorded up to four decimal points. Following are the steps performed to response the query: “Change in modularity value by DPSO and NVDBA is not statistically significant?” For answering this query, Wilcoxon Sign-Rank Test is performed on datasets listed in Table II. Firstly, the hypothesis is framed as:

Null Hypothesis H_0: There is no significant difference in modularity value obtained by DPSO and NVDBA.

Alternative Hypothesis H_1: There is a significant difference in modularity value obtained by DPSO and NVDBA.

Test statistics for all datasets are evaluated and compared using *Z* statistic and Asymp. Sig. (2-tailed) (Asymptotic Significant (2-tailed)) at 5% significance level. The Asymp. Sig. (2-tailed) is *p*-value for the test. To take the decision for accepting or rejecting the H_0, *p*-value will be used. If the *p*-value is less than specified level 0.05 then the null hypothesis H_0 will be rejected else accepted.

To understand the results of Wilcoxon Sign-Rank Test, tables viz. Ranks (Table V) and Test Statistics (Table VI) are examined.

The negative ranks, positive ranks and ties in Wilcoxon sign-Rank Test are estimated by

1) NVDBA_dataset < DPSO_dataset
2) NVDBA_dataset>DPSO_dataset
3) NVDBA_dataset=DPSO_dataset

on all the datsets by DPSO and NVDBA.

In Table V, *a-z* and *aa* reprsents negative rank, positive rank and tie respectively for each dataset listed in Table II.

From Table V, it can be viewed that in all the runs, NVDBA has obtained higher modularity value in comparison to DPSO. The Table VI has a *p*-value of less than 0.05 for all the datasets. Hence, null hypothesis H_0 is rejected which implies that the alternate hypothesis prevails i.e. change in modularity value obtained by NVDBA is statistically significant change.

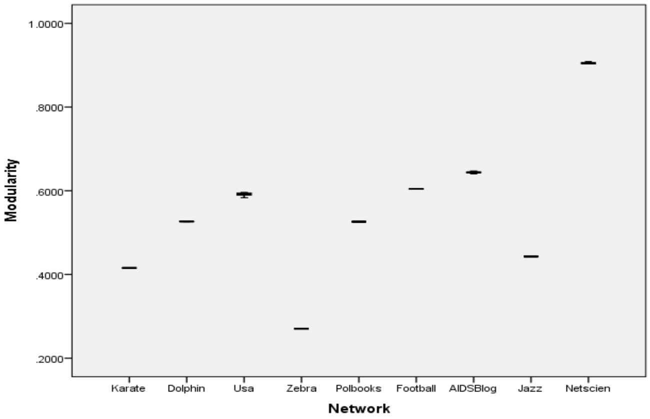


Fig. 11. Statistical Value of Modularity on Real-World Networks.

TABLE V. WILCOXON SIGN-RANK TEST: RANKS

Algorithm_Dataset		N**	Mean Rank	Sum of Ranks
NVDBA_Karate - DPSO_Karate	Negative Ranks	0[a]	.00	.00
	Positive Ranks	15[b]	8.00	120.00
	Ties	0[c]		
	Total	15		
NVDBA_Dolphin - DPSO_Dolphin	Negative Ranks	0[d]	.00	.00
	Positive Ranks	15[e]	8.00	120.00
	Ties	0[f]		
	Total	15		
NVDBA_Usa - DPSO_Usa	Negative Ranks	1[g]	1.00	1.00
	Positive Ranks	14[h]	8.50	119.00
	Ties	0[i]		
	Total	15		
NVDBA_Zebra - DPSO_Zebra	Negative Ranks	0[j]	.00	.00
	Positive Ranks	10[k]	5.50	55.00
	Ties	5[l]		
	Total	15		
NVDBA_Polbooks - DPSO_Polbooks	Negative Ranks	0[m]	.00	.00
	Positive Ranks	15[n]	8.00	120.00
	Ties	0[o]		
	Total	15		
NVDBA_Football - DPSO_Football	Negative Ranks	0[p]	.00	.00
	Positive Ranks	15[q]	8.00	120.00
	Ties	0[r]		
	Total	15		
NVDBA_AIDSBlog - DPSO_AIDSBlog	Negative Ranks	0[s]	.00	.00
	Positive Ranks	15[t]	8.00	120.00
	Ties	0[u]		
	Total	15		
NVDBA_Jazz - DPSO_Jazz	Negative Ranks	0[v]	.00	.00
	Positive Ranks	15[w]	8.00	120.00
	Ties	0[x]		
	Total	15		
NVDBA_Netscience - DPSO_Netscience	Negative Ranks	0[y]	.00	.00
	Positive Ranks	15[z]	8.00	120.00
	Ties	0[aa]		
	Total	15		

TABLE VI. WILCOXON SIGN-RANK TEST: TEST STATISTICS

Statistics	NVDBA_Karate - DPSO_Karate	NVDBA_Dolphin - DPSO_Dolphin	NVDBA_Usa - DPSO_Usa	NVDBA_Zebra - DPSO_Zebra	NVDBA_Polbooks - DPSO_Polbooks	NVDBA_Football - DPSO_Football	NVDBA_AIDSBlog - DPSO_AIDSBlog	NVDBA_Jazz - DPSO_Jazz	NVDBA_Netscience - DPSO_Netscience
Z	-3.425[b]	-3.408[b]	-3.351[b]	-2.919[b]	-3.420[b]	-3.408[b]	-3.408[b]	-3.451[b]	-3.411[b]
Asymp. Sig. (2-tailed)	.001	.001	.001	.004	.001	.001	.001	.001	.001

a. Wilcoxon Sign Rank Test

b. Based on Negative Ranks

VII. Conclusions and Future Scope

Many meta-heuristic approaches have been applied for community discovery problem. In this work, NVDBA is proposed to deal with community detection problem. The proposed algorithm NVDBA makes use of modularity as the fitness function is widely used metric. Input parameters of the proposed algorithm are initialized after a set of experiments besides maintaining the constraint on varying the loudness and pulse emission rate as per behavior of a bat. Experimental outcomes infer that the performance of NVDBA is encouraging as highest modularity is attained for almost all the datasets. Comparative analysis reveals that it produces reliable and quality community structures in comparison to LPA, DPSO, and DBA. However, still, there is the scope for improvement to suggest an improved NVDBA. The proposed method is tested on undirected and unweighted real-world networks. However, this work may further be extended to weighted and directed networks as well. In future, in-depth analysis of NVDBA can be carried out by incorporating the impact or influence of neighbors node. Such investigation is prospected to work better and yields high-quality community structures.

References

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," Proc. Natl. Acad. Sci., vol. 99, no. 12, pp. 7821–7826, 2002.

[2] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," no. 2, pp. 1–5, 2003.

[3] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," Phys. Rev. E, vol. 70, p. 66111, 2004.

[4] S. Gregory, "Finding overlapping communities in networks by label propagation," Other, vol. 12, 2010.

[5] G. Palla, I. Farkas, I. Derényi, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society.," Nature, vol. 435, no. 7043, pp. 814–8, 2005.

[6] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech. Theory Exp., vol. 2008, no. 10, pp. 1–12, 2008.

[7] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," pp. 1–12, 2007.

[8] B. M. Brentan, E. Campbell, G. L. Meirelles, E. Luvizotto, and J. Izquierdo, "Social Network Community Detection for DMA Creation: Criteria Analysis through Multilevel Optimization," Math. Probl. Eng., vol. 2017, 2017.

[9] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5199 LNCS, pp. 1081–1090, 2008.

[10] Z. Shi, Y. Liu, and J. Liang, "PSO-Based Community Detection in Complex Networks," 2009 Second Int. Symp. Knowl. Acquis. Model., pp. 114–119, 2009.

[11] S. Sadi, "Community Detection Using Ant Colony Optimization Techniques," 15th Int. Conf. Soft Comput. MENDEL 2009, pp. 206–213, 2009.

[12] E. Ali, H. Hussein, A. Hafez, A. E. Hassanien, and A. Fahmy, "A Discrete Bat Algorithm for the Community Detection Problem," in Hybrid Artificial Intelligent Systems, 2015, vol. 9121, no. June, pp. 188–199.

[13] W. Chunyu and P. Yun, "Discrete Bat Algorithm and Application in Community Detection," pp. 967–972, 2015.

[14] A. Song, M. Li, X. Ding, W. Cao, and K. Pu, "Community Detection Using Discrete Bat Algorithm," Int. J. Comput. Sci., no. February, 2016.

[15] X. S. Yang, "A new metaheuristic Bat-inspired Algorithm," Stud. Comput. Intell., vol. 284, pp. 65–74, 2010.

[16] S. Mirjalili, S. M. Mirjalili, and X. S. Yang, "Binary bat algorithm," Neural Comput. Appl., pp. 1–19, 2013.

[17] Q. Cai, M. Gong, B. Shen, L. Ma, and L. Jiao, "Discrete particle swarm optimization for identifying community structures in signed social networks," Neural Networks, vol. 58, pp. 4–13, 2014.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," Neural Networks, 1995. Proceedings., IEEE Int. Conf., vol. 4, pp. 1942–1948 vol.4, 1995.

[19] V. M. Kirkpatrick S, Gelatt CD, "Optimization by simulated annealing," Science (80-.)., vol. 220, no. 4598, pp. 671–680, 1983.

[20] M. E. J. Newman, "Detecting community structure in networks," Eur. Phys. J. B, vol. 38, no. 2, pp. 321–330, 2004.

[21] I. F. J. A, I. Technologies, and C. Science, "A comprehensive review of bat algorithms and their hybridization by," no. September, 2013.

[22] H. D. Abatari, M. S. S. Abad, and H. Seifi, "Power Flow," pp. 793–798, 2016.

[23] W. W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups Author (s): Wayne W . Zachary Published by : The University of Chicago Press Stable URL : http://www.jstor.org/stable/3629752," J. Anthropol. Res., vol. 33, no. 4, pp. 452–473, 1977.

[24] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: Can geographic isolation explain this unique trait?," Behav. Ecol. Sociobiol., vol. 54, no. 4, pp. 396–405, 2003.

[25] D. E. Knuth, The Art of Computer Programming, Volume 4, Fascicle 0: Introduction to Combinatorial and Boolean Functions. Addison-Wesley, 2008.

[26] S. R. Sundaresan, I. R. Fischhoff, J. Dushoff, and D. I. Rubenstein, "Network metrics reveal differences in social organization between two fission-fusion species, Grevy's zebra and onager," Oecologia, vol. 151, no. 1, pp. 140–149, 2007.

[27] "V. Krebs," Unpublished. [Online]. Available: http://www.orgnet.com/.

[28] S. Gopal, "The evolving social geography of blogs," in Societies and CIties in the Age of Instant Access, E. H. Miller, Ed. Berlin:Springer, 2007, pp. 275–294.

[29] P. Gleiser and L. Danon, "Community Structure in Jazz," pp. 1–12, 2003.

[30] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys., vol. 74, no. 3, pp. 1–19, 2006.

[31] S. Garc, D. Molina, M. Lozano, and F. Herrera, "A Study on the Use of Non-Parametric Test for Analyzing the Evolutionary Algorithms ' Behaviour : A Case Study on the CEC ' 2005 Special Session on Real Parameter Optimization A Study on the use of Non-Parametric Test for Analyzing the Evolutionary Algori," 2005.