Applied Intelligence (2019) 49:2119–2136
https://doi.org/10.1007/s10489-018-1386-9

# A multi-objective bat algorithm for community detection on dynamic social networks

Imane Messaoudi[1] · Nadjet Kamel[2]

Published online: 2 January 2019

**Abstract**
Many evolutionary algorithms have been proposed to deal with the problem of community detection in social dynamic networks. Some algorithms need to fix parameters in advance; others use a random process to generate the initial population and to apply the algorithm operators. These drawbacks increase the search space and cause a high spatial and temporary complexity. To overcome these weaknesses, we propose in this paper a novel multi-objective Bat Algorithm that uses Mean Shift algorithm to generate the initial population, to obtain solutions of high quality. In our proposal, Bat Algorithm simultaneously optimizes the modularity density and the normalized mutual information of the solutions as objective functions. The operators of the algorithm are applied to the problem of community detection in social dynamic networks by giving another sense to the velocity, frequency, loudness and the pulse rate of natural Bat. The algorithm keeps the principal of the Mean Shift algorithm to generate new solution and avoid the random process by defining a new mutation operator. The algorithm does not need to the non-dominated sorted approach or the crowding distance, but it attributes a weight to each objective function. The method is tested on artificial and real dynamic networks and the experiments show satisfactory results in terms of normalized mutual information, modularity and error rate.

**Keywords** Community detection · Modularity density · Mean shift · Bat algorithm · Dynamic network

## 1 Introduction

In social networks, the increasing number of users and interactions between them imply a complex structure of the system which needs to be analyzed. In the literature, complex systems have the capability of being represented as graphs. This representation facilitates the comprehension of their structure and allows to the revealing of some important information. Taking the example of social network, which is a set of individuals connected with different relationships. The structure of such network can be represented as graph where the nodes design the individuals and the links design the relations between them. It is known that individuals in social networks organize themselves into communities, where the nodes of the same community are very dense and close compared to the rest of the network.

Another major characteristic of social networks is the dynamic evolution, where we can define a dynamic network as a network that changes its structure (nodes and interactions) over the time. In social network analysis, community detection is one of the main tools that play an important role as revealing new relations and new information. But due to the dynamic evolution of social networks, community detection becomes more and more significant problem, since it can reveal the transformations that a network can undergo over the time.

Many research works have been done to deal with the problem of community detection on static networks, such as label propagation SLPA in [1], clustering algorithm in [2], genetic algorithm in [2], bee swarm optimization in [3] and hybrid optimization algorithm in [4]...etc. However the evolution of the network over time implies a lot of changes as the addition of new nodes or the removal of edges. For example, in Facebook, at any time an individual can add friends; can be member of new groups ...etc. For these reasons, community detection algorithms in static networks cannot be applied directly on dynamic networks.

✉ Imane Messaoudi
imessaoudi@usthb.dz

Nadjet Kamel
nkamel@univ-setif.dz

[1] LRIA Laboratory, Department of Computer Science, Universite de Sciences et de la Technology Houari Boumediene, BP 32 El alia 16111 Bab ezzouar, Algiers, Algeria

[2] Faculty of Sciences, Department of Computer Science, Université Ferhat Abbas Setif 1, Setif, Algeria

Thus, several research works are interested to analyze dynamic networks which are classified into two categories "non-evolutionary" category and "evolutionary category". In the non-evolutionary category, or the two-stage category the proposed algorithms do not take the historical community detection into account. They treat the network structure independently of its evolution over the time, such as the algorithms proposed in [5] and the algorithm proposed in [6].

To overcome this weakness, the evolutionary category of algorithms appears where we can cite the algorithm proposed in [7]. The algorithm is an evolutionary link community detection based on the weights of edges to cluster the nodes. At each time step the algorithm needs to the parameter *alpha* to control the size of the detected communities. We can also cite FacetNet proposed by [8]. FacetNet is a framework for analyzing communities and their evolution in dynamic networks. According to [6], the algorithm allows the participation of individuals in multiple communities at the same time and with different participation levels. However, only link information is considered and the model is only used for explaining the observed data, it is not possible to predict the future behavior of the individuals of a network. The algorithm also needs to define the values of some parameters such as the number of communities.

Furthermore, the multi-objective optimization algorithms have been proposed. These algorithms do not need to fix parameters in advance and take the historical community structure into account by using a cost function. This function is composed of the snapshot cost (SC) which measures the quality of the community structure at time *t*, and the temporal quality (TC) which evaluates the similarity between the community structure at time *t* and *t*-1. The algorithm proposed in [9] is a multi-objective evolutionary algorithm for detecting communities in dynamic networks using genetic algorithm. Another multi-objective approach is proposed in [10] using biogeography metaheuristic.

The major drawback of the multi-objective evolutionary algorithms is the random generation of initial population that increases the search space and causes a high spatial and temporary complexity. The method presented in [11] is an evolutionary computing approach for dynamic community detection based on label propagation algorithm to generate the initial population of the evolutionary algorithm. Although, the opinion presented in [12] says that "the label propagation method has the shortcomings of uncertainty and randomness in the label propagation process." This last affects the accuracy and the stability of the communities.

Thus, in this paper we propose a multi-objective Bat Algorithm (BA) based on Mean Shift (MS) approach. BA takes the advantage of echolocation of microbats, adjusts the frequency and velocity automatically and increases the pulse rate when it is close to the best solution. BA also combines the advantages of PSO and GA and it is superior to them in terms of accuracy and efficiency. In order to resolve the problem of premature convergence and reduce the search space, we use the MS approach to generate the initial population. The MS approach returns a set of high quality solutions and has the advantage of running fast with minimal error. MS also does not need to fix the number of clusters in advance. In addition we derive the principal mechanism of MS algorithm to define a new mutation and get high quality solutions. This mutation adopts the principal of MS algorithm to escape from the low quality of random traditional mutation.

The algorithm uses two objective functions, the modularity density to maximize the snapshot quality and the normalized mutual information to minimization of temporal cost. In addition, the multi-objective optimization is efficient than a single objective. Because it returns a set of non-dominated optimal solutions compared to the single objective which returns one solution. The method is tested on real and artificial networks and the obtained results are satisfactory compared to other methods.

The main contributions of our proposal are:

- The utilization of MS approach to generate the initial population provides high quality clustering.
- The utilization of multi-objective BA to optimize two objective functions simultaneously.
- A new solution is generated by adjusting frequency and velocity. A new mutation is defined based on the MS principal mechanism.
- The complexity of the proposed is polynomial.
- The paper is organized as follow: in Section 2 we detail the related works and in Section 3 we give the preliminaries of the work. Section 4 explains the method and Section 5 is devoted to the experiments. Finally we give the conclusion in Section 6.

## 2 Related works

Community detection is a fundamental concept that allows us to understand the main structure of the networks. In static networks, a large number of methods have been proposed such as the algorithm proposed in [13]. This algorithm divides the nodes of the network into several communities; next it merges them to increase the value of modularity. Finally, it returns hierarchical representation called "dendrogram" which is cut on the level that have optimized the value of modularity. Another algorithm presented in [14] deals with the problem of community detection. This algorithm runs on two steps. In the first step, each node of the network is put in its community, next the communities are merged by optimizing the gain in terms of modularity. The second step considers the communities of the first step as nodes of a new network. The two phases are repeated till no optimized value of modularity remains.

The work presented in [15] uses a random walk strategy to decompose the network into several communities. In this algorithm an information theoretic approach is employed where the information flows in the real system are seen as the probability flows of random walks. The algorithm is applied on directed and weighted networks. The label propagation strategy is also used in the process of detecting communities in networks. The paper presented in [16] initializes each node of the network with a unified label. In next step, each node takes the label that occurs on its neighbors. Finally, the nodes with a same label are considered as one community. In [2] a genetic algorithm is proposed to deal with the problem of community detection. The algorithm optimizes several objective functions simultaneously, expressed by the combination of the Density, Centralization, Heterogeneity, Neighbourhood, and Clustering Coefficient in one objective function. The population of the algorithm is represented as binary vectors. The algorithm uses a random crossover and mutation points.

The work proposed in [17], is a multi-objective genetic algorithm applied on the problem of community detection. The algorithm uses the locus-based adjacency representation to represent the population which is generated randomly. The algorithm uses crossover, mutation and selection operators to maximize the intra-connections inside the community and minimize the inter-connections between communities. Ant colony optimization is also applied on the problem of community detection such as the algorithm proposed in [18]. The algorithm simulates the behavior of real ants to the nodes clustering in a network. Each node in the network is represented as ant in a virtual grid. One ant moves in the grid by calculating a probability and a pheromone diffusion model is proposed where the ant colony movement determines the network partitioning.

From the works cited above, we can see that some of them use exact methods to detect communities in social networks and others use evolutionary methods. However, the dynamic evolution of real social networks makes the analysis of static networks non-significant. In addition, these approaches do not work well on dynamic networks. For this reason many works concentrate on the analysis of dynamic networks.

The paper proposed in [6] presents a survey on community detection in dynamic networks. The work classifies the methods into several categories which are: Event based methods, Random walk methods, Probabilistic methods; Modularity based methods and Evolutionary methods. In this paper, we classify them into evolutionary-based methods and non-evolutionary-based methods.

## 2.1 Non-evolutionary-based methods

In this section, we can enumerate several approaches beginning by iLCD suggested in [5] which uses k-clique clustering to propose an incremental k-clique clustering method. In this work the dynamics of social networks is represented as change stream, the depth first search is used to solve the incremental k-clique clustering. However, the clique percolation based method has worst results on large networks. In addition, the incremental spectral clustering is sensitive to scale parameters and need to several improvements such as the improvement of creating similarity matrix and dealing with large-scale networks as mentioned in [19].

A local dynamic method is proposed in [20], it concentrates on the part of network that changes over time. At iteration $t = 0$, a static algorithm for community detection is executed and then a PageRank-approach is performed which needs to the seeds. These seeds are selected using a seeding strategy to track the behaviors of dynamic communities which construct a partial evolution graph. It is known that the PageRank approach run fast. However the seeding strategy applied to found good seeds, performs a lot of test on each node to check if it is a local-minimal conductance node or not, which is not practical. In [21] NEIWalk is suggested, the work modifies a content-based network to a node and edge interaction and it integrates the linkage structure, node content and the edge content. The major drawback of this work is a bounded accuracy loss caused by the random walk sampling.

In [22] a community event prediction on dynamic network is suggested. The framework can be summarized in four steps: community detection, feature extraction, community matching and finally the phase of classification. This last step considers the events such as: growth, merge, split ...etc. as classes. In the phase of community matching, a Jaccard coefficient is calculated between two consecutive snapshots to verify if the two communities match each other. Consequently, the corresponding event is determined on the basis of three matching thresholds values. So the disadvantage of this approach is the requirement of fixing the values of these thresholds.

The work presented in [23] is an event-based framework which uses an incremental algorithm to detect the evolution of event in the interaction graph. In this approach a measure of stability, sociability is computed. According to [6], the framework is based on the use of certain critical events that facilitate the ability to compute and reason about the novel behavior oriented measures. This last, can offer new and interesting insights for the characterization of dynamic behavior of such interaction graphs. The remarkable inconvenient is the ignoring of historic community structure due to the clustering of each snapshots graph independently.

## 2.2 Evolutionary-based methods

In [8] Facetnet is proposed. It is a framework for analyzing communities and their evolutions in dynamic networks. The work represents the problem as non-negative matrix factorization. The framework also uses a soft clustering algorithm from static graph to dynamic and then performs an iterative algorithm in order to converge to the optimal solutions. But the method needs to fix the number of communities in advance and cannot approximate well the ground truth which is expressed by the lower values of normalized mutual information. An evolutionary link community structure discovery in dynamic weighted networks is proposed in [7]. The algorithm is based on the weights of edges to cluster the communities and it needs to the parameter *alpha* to control the size of the communities at each time step.

The algorithm ESPRA, proposed in [24], extends the perturbation theory in quantum physics for static networks to dynamic networks. In the algorithm a new similarity including structural perturbation and topology is proposed, and the temporal smoothness framework is employed as evolutionary clustering algorithm. This framework is defined as a linear combination of the similarity on dynamic network. The process of the algorithm begins by choosing the centers of clusters that have high density, and then classifies the nodes. In second step, a modified modularity is used to merge the clusters. The drawbacks of the algorithm are a complicated process of structural perturbation and the need of the parameter $\beta$ to define the combined similarity. In [25] a multi-objective framework for community detection is proposed. The authors of this work conclude that the major evolutionary methods for community detection on dynamic network have a same principle which is the optimization of snapshot cost and temporal cost.

DYNMOGA is a multi-objective evolutionary algorithm to detect communities in dynamic networks proposed in [9]. The work is a multi-objective genetic algorithm formulated to detect communities with temporal smoothness. The advantage of the algorithm is that it automatically returns a solution representing the best trade-off between the accuracy of the obtained clustering, and the deviation from one time step to the successive one. The weakness of the algorithm is the high execution time produced by the random generation of the initial population.

Another multi-objective approach is proposed in [10], the work presents a multi-objective biogeography based optimization algorithm which uses a decomposition mechanism to detect the communities. This mechanism optimizes normalized mutual information and modularity. In addition two operators are designed to improve the effectiveness of the algorithm which are the problem-specific migration and mutation. The work also suffers from the problem of generating the initial habitat population randomly.

DYN-MDPSO proposed in [26] is a multi-objective particle swarm optimization which maximizes the modularity density and the normalized mutual information simultaneously. The algorithm uses the decomposition framework, meaning that the algorithm is decomposed into single objective sub-problems. The algorithm also uses new updating rules to generate new solutions.

The major drawback of the multi-objective approaches is the random generation of initial population which increases the search space and causes high spatial and temporary complexity. The method presented in [11] is an evolutionary computing approach for dynamic community detection. The initial population of this evolutionary algorithm is generated on the basis of label propagation approach. But according to the opinion of [12], the label propagation method has the shortcomings of uncertainty and randomness in the label propagation process, which affects the accuracy and stability of the community.

From the above studies we can conclude that there are several static community detection algorithms which cannot be applied on dynamic networks. In dynamic community detection, the non-evolutionary methods need to fix some parameters in advance as [22], and do not take into account the historic community structure such as [23]. However, the evolutionary methods take a lot of time and iterations to converge due to the random initialization of population as DYN-MOGA [9]. The work of [11] uses a label propagation process to get high quality solutions, but the label propagation also has a percentage of uncertainly. The work also uses a genetic algorithm to optimize two objective functions, where it performs in several iterations with crossover, mutation and selection operators, and it sorts the solutions with fast non-dominated sorted approach. In this paper, we propose a multi-objective BA based on MS to have solutions of high quality. The algorithm optimizes two objective functions and assigns to each one a weight.

According to [27], the multi-objective optimization methods are classified into interactive methods, posteriori methods, and a priori methods, depending on the decision maker (DM) preferences. We can add also the no-preference methods where the preferences of the decision maker are not defined, but at the end of the process, the DM accepts or rejects the solutions.

1. Interactive methods: In this field the process of decision and optimization are combined such as the DM defines the preferences and modifies them when the optimization process runs.
2. Posteriori methods: In these methods the preference of the DM is not required. However, the DM takes the set of good solution from the generated pareto solutions.

3. A priori methods: In this field the opinion of the DM is required before starting the optimization process. The defined preferences help to search for the pareto optimal solutions that match it. The DM also knows the priority of the objective functions which will be transformed into a single one.

   Many approaches are used to combine the objective functions and resolve the problem as mono-objective such as:

   3.1 The ε-constraint approach: This class uses one objective function and considers the others as constraints.

   3.2 The goal programming: in this class the DM must define the goals $T_i$ of all objectives $f_i$. The new objective function is defined such as it decreases the difference between the goals and the results.

   3.3 The weighted sum methods: in this class field the weights are attributed to each objective function, such as the sum of all weights is equal to one; expressed as:

$$(\textbf{Minimize})\sum_{i=1}^{k} w_i f_i^s(X) \quad w_i \geq 0 \quad \sum_{i=1}^{k} w_i = 1$$

Because our algorithm uses weights to combine the two objective functions we classify it as a priori multi-objective optimization algorithm.

Several posteriori multi-objective Bat algorithms have been proposed to deal with the problem of multi-objective optimization such as the algorithm proposed in [28]. This algorithm is a multi-objective Shuffled Bat algorithm for optimal placement and sizing of multi distributed generations in radial distribution systems. Two objective functions are minimized by using Bat algorithm as an optimization algorithm. The method does not attribute a weight to each objective function as the priori-based methods, but it chooses the best solutions from the pareto front where the non-dominated solutions are analysed with NSGA-II. However, the process is complicated and the algorithm needs to a decision variable **d** and the viable objective area **Z**.

In [29] the multi-objective optimization using Bat algorithm for shell and tube heat exchanges is proposed. The algorithm does not combine the two objective functions which are the shell and the heat exchange, but searches for a set of optimal pareto front solutions. However the optimal solutions are chosen by fixing the tube sheet and the segmental baffles.

The multi-objective Bat algorithm is also used in the design of power filter in [30]. The method uses external archive to store the non-dominated solutions and the search space of variables is divided into many hyper-cubes. The method does not combine the objective functions but need to define the constraints of each one. In [31] a multi-objective hybrid Bat algorithm for combined economic/emission dispatch is proposed. The algorithm uses the elitist strategy with external archive. The algorithm also modifies the update of velocity and the random walk of the standard Bat algorithm. In order to have a distributed pareto optimal front a modified crowding distance is employed.

From the works cited above, we can conclude that the posterior Multi-objective optimization algorithms have complicated process and need some parameters in some cases. However, the process in priori based-methods is simple and the affectation of weights facilitates the sorting of solutions. For these reasons we choose a priori method in our proposal. In addition, BA uses the velocity and frequency to generate a new solution and control the convergence by increasing the pulse rate when it closes to the best solution. The new defined mutation based on MS approach, also increases the efficiency of the algorithm, because it is guided and it is different from the random traditional mutation. At the best of our knowledge neither MS algorithm nor BA algorithm have been applied on the problem of community detection in dynamic network. This motivates us to hybridize the two approaches to have benefits from their advantages.

## 3 Preliminaries

### 3.1 Bat inspired algorithm

BA is a bio-inspired algorithm based on swarm intelligence. The algorithm was developed by Xin She in 2010. As defined in [32], BA follows echolocation of Bats by using sonar echoes to detect and avoid obstacles. It is generally known that sound pulses are transformed into frequencies which are reflected from obstacles. Xin-She in [32] proposed three generalized rules for BA:

- All Bats use echolocation to sense distance, and they also guess the difference between food/prey and background barriers in some magical way.
- Bats fly randomly with velocity *vi* at position *xi* with a fixed frequency *fmin*, varying wavelength and loudness *A*0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission *r* depending on the proximity of their aim.
- Although the loudness can vary in many ways. It is assumed that the loudness varies from a large (positive) *A*0 to a minimum constant value *Amin*.

The application areas of Bat algorithm are several such as: the detection of intrusion in [33], the association rule mining in [34] and the mining of numerical association rules in [35]…etc.

## 3.2 Multi-objective optimization problem

The multi-objective optimization problem can be defined as a simple optimization problem which includes at least two objective functions, expressed as suggested in [36] by:

$$\min\left(F(x) = [F_1(x); F_2(x); \ldots; F_k(x)]^T\right) \quad (1)$$

Subject to $g_i(x) < = 0$ j = 1; 2… m and $hl(x) = 0$ $l = 1; 2; \ldots; e$.

Where $k$ is the number of objective functions, $m$ is the number of inequality constraints, and $e$ is the number of equality constraints. $x \in E^n$ is a vector of decision variables. $F(x)$ is a vector of objective functions.

**Definition 1** The feasible design space X:

$X$ is defined as the set$\{x| g_j(x) < = 0\}$, j = 1; 2; …; $m$ and $hi(x) = 0$, $i = 1, 2 \ldots, e$.

**Definition 2** The feasible criterion space Z is a set of $\{F(x)| x \in X.\}$

**Definition 3** Non-Dominated and Dominated Points:

A vector of objective functions, $F(x^*) \in Z$, is non-dominated iff there does not exist another vector, $F(x) \in Z$ such that $F(x) < = F(x^*)$ with at least one $Fi(x) < F_i(x^*)$, otherwise, $F(x^*)$ is dominated.

**Definition 4** *Pareto Optimality*

The Pareto front PF of a multi-objective can be defined as the set of non-dominated solutions so that: $PF = \{x \in F | \neg \exists x' \in F : f(x') < f(x)\}$

### 3.2.1 Multi-objective bat algorithm

As mentioned in [37] the multi-objective optimization problems are more complicated than the single objective optimization because it should find and/or approximate the optimality fronts. The pseudo code of the multi-objective BA taken from [37] is as follow:

**Algorithm 1 Multi-objective Bat Algorithm**

```
1: Objective functions $f_1(x); \ldots ; f_k(x)$, x=( $x_1, \ldots, x_d)^T$
2: Initialize the bat population $x_i$ (i = 1, 2, ..., n) and $v_i$
3:
4: for (j=1 to N) do (points on pareto fronts)
5:     Generate $k$ weights $w_k$ >= 0 so that $\sum_{k=1}^{K} w_k = 1$
6:     Form a single objective $f = \sum_{k=1}^{K} w_k f_k$
7:
8:    while t < Max number of iteration do
9:      Generate new solutions by adjusting frequency,
10:     and update velocities and locations/solutions
11:
12:     if rand > $r_i$ then
13:       Random walk around a selected best solution
14:     end if
15:     Generate a new solution by flying randomly
16:
17:     if rand < $A_i$ and $f(x_i) < f(x*)$ then
18:       Accept the new solutions
19:       Increase $r_i$ and reduce $A_i$
20:     end if
21:     Rank the bats and find the current x*
22:  end while
21:  Record x* as non-dominated solution
23: end for
24: Postprocess results and visualization
```

In the lines 1 and 2 the initialization of the objective functions and the bat population is done.

In the lines 5 and 6 ***k*** weights are affected to each objective function, such as the sum of weights is equal to one.

In line 8: while the maximum number of iterations is not meet new solutions are generated by adjusting frequency and updating velocity.

From line 12 to 14 if a random number is less than the pulse rate a random walk around the best solution is done.

From line 17 to 20: if a random number is less than the loudness and the objective function of the current solution is less than the best one, the new solution is accepted, increase ***r*** and reduce ***A***. At the end of the iteration the current best solution is found.

## 3.3 Mean shift algorithm

Mean Shift algorithm is a kind of clustering data which does not need to fix any parameter in advance. The algorithm was developed in 2002 [38]. In this paper we apply this algorithm to the problem of community detection, as its idea is very simple and its results are improved. In addition the algorithm does not need to fix the number of clusters as the case of k-means algorithm. The algorithm affects the points of data iteratively to the clusters that have the highest density of datapoints. The process of the algorithm begins by calculating the neighbors $N(x)$ for each point $x$ in the set of datapoints $X$, by the Euclidean distance, and then calculates the mean shift $m(x)$ by the following equation:

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)} \quad (2)$$

Where $K$ is the Gaussian Kernel. In next step all $x \in X$ are changed to $m(x)$. The process is repeated a number of iterations until convergence or stability.

# 4 Proposed approach

This section shows the details of the method adopted in this paper, which uses BA to detect the communities of the network. The initial population of the algorithm is generated according to the sub-section 4.1. Each bat in the generated population is represented according to 4.2. These bats are evaluated by using two objective functions defined in sub-section 4.3. The proposed mutation is defined in 4.4. After that, we follow the behavior of bat defined in sub-section 4.5 to execute the method described in the flowchart of Fig. 3.

### 4.1 Initial mean shift algorithm

To get the preliminary partition with Mean Shift we need to use the similarity matrix as input. This matrix is obtained by converting the $n$ nodes network to a signaling process which is done as indicated in [39, 40] as follow:

First, the identity matrix is added to the adjacency matrix of the network: $Z=(In+A)^t$. After that, each element of the matrix is normalized by dividing it by the root of the sum of the other elements of the squared matrix.

In next step, the Euclidean distance between the vectors of the obtained matrix is calculated to serve as input of the MS approach described in the section 3.3. By varying the number of iterations in the algorithm and the bandwidth of the Gaussian Kernel, we return the maximum number of solutions with high quality.

### 4.2 Representation of solutions

In the encoding step, the individuals (bats) are represented by the so called "string-based representation" noted as $\mathbf{B}=[b1, b2, \ldots, bi,. .., bn]$ where $bi$ corresponds to the label of the community to which the node $i$ belongs. $n$ is the number of nodes in the network. Table 1 corresponds to the encoding of the network of Fig. 1.

All nodes with the same label should be in the same community. For example, the nodes 2 and 3 have the label 1, and should be in the community 1.

### 4.3 Objective functions

In order to evaluate the solutions we need to use the objective function. In this paper we adopt two objective functions to formulate a multi-objective algorithm. The function widely used in community detection problems is the modularity density D defined in [41] as:

$$D=\sum_{i=1}^{N}\frac{L(V_i,V_i)-L\left(V_i,\overline{V}_i\right)}{|V_i|} \tag{3}$$

**Table 1** Example of bat's encoding

| Node i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Bat **B** | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |

Where $L(V_i, V_i)$ is the number of links in the community $V_i$, $L\left(V_i,\overline{V}_i\right)$ is the number of links with one node in the community Vi and other node outside $V_i$. | $V_i$| is the number of nodes in the community $V_i$.

The second objective function is the normalized mutual information defined in [42] as a measure criterion adopted to know if the true partition of the network and the detected one are similar or not. In this paper we calculate the normalized mutual information between the partition at time t and t - 1.

### 4.4 The mutation operator

In this section we define the mutation operator derived from MS algorithm. We begin by searching the neighbors of the nodes in each community of the partition. Second, we compute the mean shift $m(x)$ between the corresponding node and its neighbors. Next, we calculate the Euclidean distance between the node and the $m(x)$. Finally, the node takes the label of the neighbors that have minimum distance, as showed in Algorithm 2.

**Algorithm 2**: The mutation operator

```
Input: The node
P= partition(network)
for all neighbors in P do
    //calculates its mean shift
    $m_i$=mean_shift(the node, neighbors)
end for
l=min(Euclidiean_distance(the node, $m_i$)
label (the node)= l
```

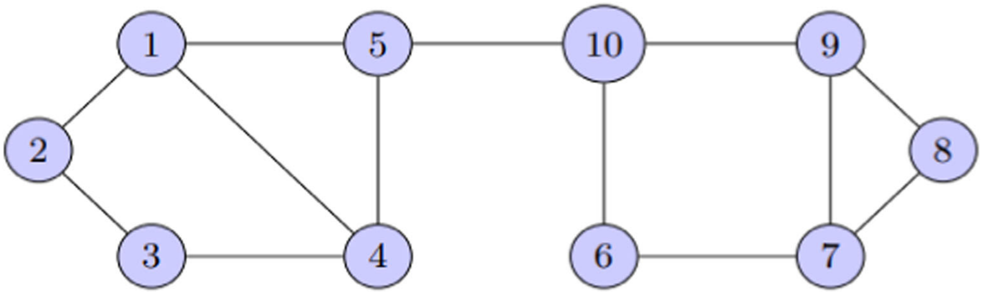


**Fig. 1** Sample network

## 4.5 Behavior of bats

Naturally, bat sends a signal with a frequency *f*. This signal is sent back as echo while the bat finds its object. We simulate this movement by the artificial bats.

In this paper we consider the velocity *vi* as a set of permutations that allows each bat to change its position and to create another two bats by improving the objective functions. The frequency *fi* corresponds to the positions chosen on the bat where the modification will be done. The frequency is calculated by the following equation:

$$f_i = (1+f)\beta \qquad modulo\,[f] \qquad \beta\in[0,1] \tag{4}$$

Where *f*, refers to the length of the solution.

The new solution is transformed according to Fig. 2.

In Fig. 2 we calculate two frequencies *fmin* and *fmax* on two individuals. By adding the velocity to the solutions we obtain four children. Finally, if child1 dominates child2, we take child1 otherwise we take child2. If child3 dominates child4 we take child3, otherwise we take child4. The flowchart of our modified Bat Algorithm is shown in Fig. 3.

The algorithm begins by the initialization of the population using MS approach.

The frequency is described by the eq. (4) and the velocity corresponds to the set of permutations that improves individuals. The position *xi* refers to the solution after defining a new velocity and a new frequency, *r*0 and *Ai* are chosen in the ranges [0, 1] and [2, 17] respectively. While *t* is less than the maximum number of iterations, we generate new solutions by adjusting *f* and *v*. In order to benefit from the quality of individuals, if a random number is superior to *ri*, we add a generated population with MS to the current population. If a random number is less than *Ai* and the objective function of the current solution is less than the best one, we mutate the best solution. We rank the bats and found the best; we increase *ri* and reduce *Ai* by applying the two following equations: $A_i^{t+1} = \alpha A_i^t$ and $r_i^t = r_i^0[1-exp(-\gamma t)]$, $\alpha=\gamma=0.9$. Where we set $A_i^0 = 2$ and $r_i^0 = 0.1$. Finally we rank the bats and return non-dominated set.

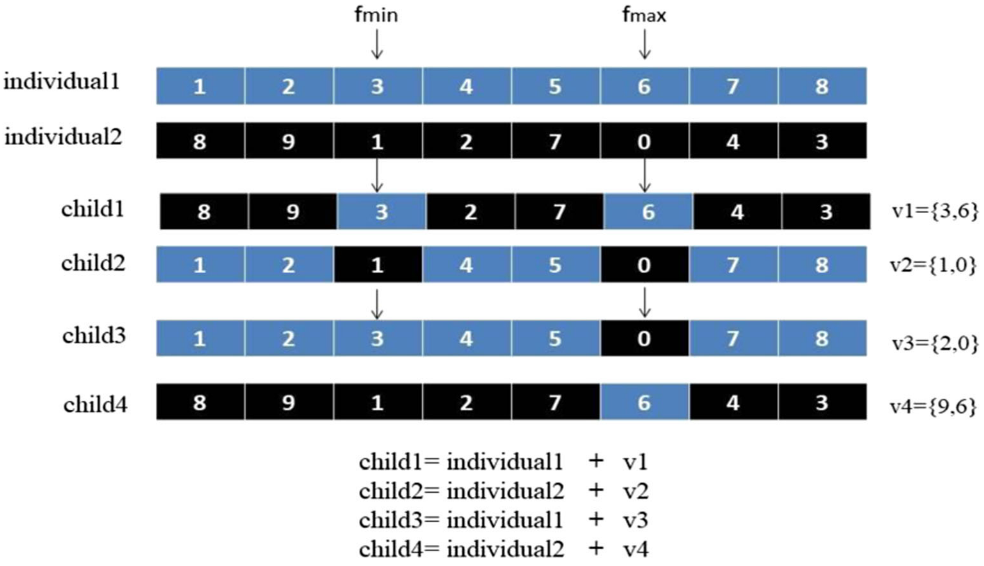
## 4.6 Complexity

The complexity of the proposed approach can be calculated as the sum of the complexity of MS algorithm and the multi-objective BA. The complexity of MS is calculated according to the size of the used identity matrix which reflects the number of node in the network and the number of iterations that the algorithm takes to converge. So, this complexity is given by complexity(MS) = *size*(*I*) $*$ *nb-iter*. The complexity of multi-objective BA is calculated according to the number of bats *k*, the number of iterations *iter* and the complexity of creating a new solution.

So complexity(multi-bat) = $k * iter*$ complexity (createnew sol), where complexity(create-new-sol) is calculated as: because we brows two solution to create two other best solutions. So the complexity is given by complexity(create-new-sol) = 2 $*$ *len*(*solution*) + *N*, where *N* is the complexity of the mutation operator: because we brows all the neighbors of a partition, and in the worst case, all the neighbors is equal to the number of nodes in the network.

This complexity is low than the complexity of NSGAII, MOPSO which is of M*n$^2$ and the complexity of MOEA which is of m*N*T, where T is the number of weight vectors in the neighborhood.

**Fig. 2** Bat's updating position



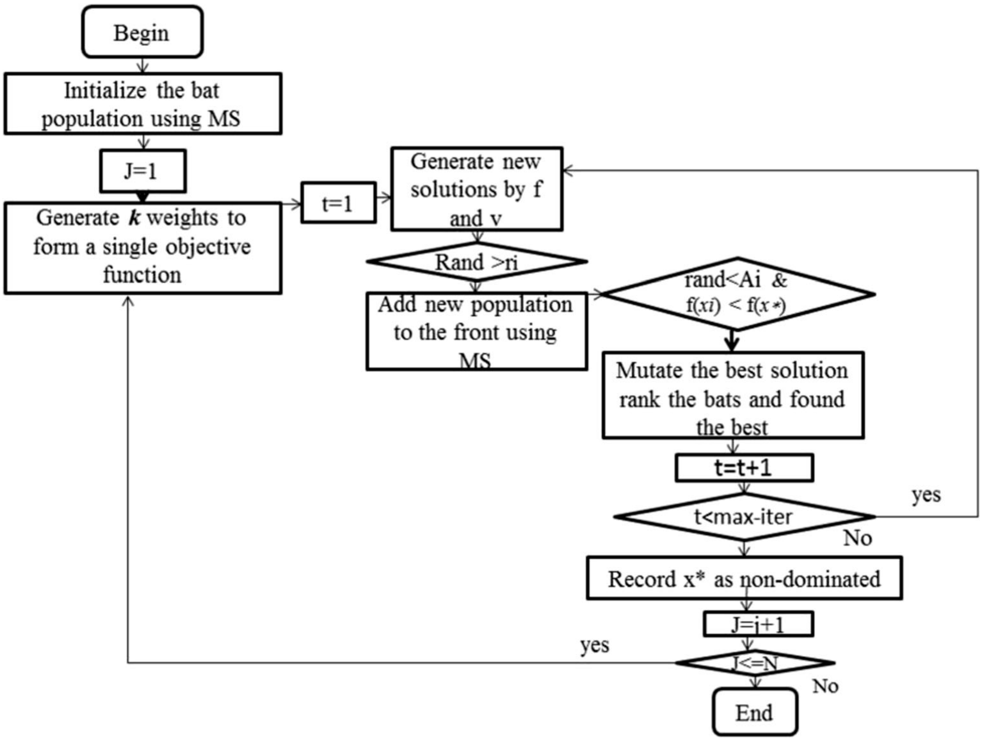

Begin
Initialize the bat population using MS
J=1
Generate k weights to form a single objective function
t=1
Generate new solutions by f and v
Rand >ri
Add new population to the front using MS
rand<Ai & f(xi) < f(x*)
Mutate the best solution rank the bats and found the best
t=t+1
t<max-iter
yes
No
Record x* as non-dominated
J=j+1
J<=N
yes
No
End

**Fig. 3** Flowchart of the modified multi-objective Bat Algorithm

## 5 Experiments

Our proposal is implemented in Python on a computer of 2.50 GHZ processor and 12G Ram. We show the performance of our method on synthetic and real networks and we compare the obtained results to the results of L-DMGA [11], DYNMOGA [43], FacetNet [8], DYN-MODPSO [26], ESPRA [24] and IsoFdp [44].

### 5.1 Evaluation criteria

To evaluate the quality of the obtained results we use the normalized mutual information (NMI), error rate defined in [42], and the modularity as metrics of evaluation.

#### 5.1.1 Normalized mutual information

As defined in [42], NMI is a measure criterion adopted to know if the true partition of the network and the detected one are similar or not. Given a network of *N* nodes, a true partition *A* and a detected partition *B*, each element *Cij* of a confusion matrix *C* corresponds to the common nodes between the community *i* and *j* of partition A and B, respectively. The measure is defined as follows:

$$NMI(A,B)=\frac{-2\sum_{i=1}^{c_a}\sum_{j=1}^{c_b}C_{ij}log\left(C_{ij}N/\left(C_iC_j\right)\right)}{\sum_{i=1}^{c_a}\left(C_ilog(C_i/N)\right)+\sum_{j=1}^{c_b}\left(C_jlog\left(C_j/N\right)\right)} \tag{5}$$

Where *Ca*, *Cb* are the numbers of the communities in the partitions *A* and *B* respectively. *Ci*, *Cj* are the numbers of nodes in the communities *i* and *j* respectively. If NMI (A, B) = 1, then the two partitions are identical, else if NMI(A,B) = 0, then the two partitions are different.

#### 5.1.2 Error rate

The error rate measures the distance between the detected communities and the real communities. To calculate it, we create an *n*∗ *k* indicator matrix *Z*, where *n* is the number of nodes and *k* is the number of communities, and a similar indicator matrix *G* to represent the real community. So

$$error(Z,G)=\left\|ZZ^T-GG^T\right\| \tag{6}$$

#### 5.1.3 Modularity

The function widely used in community detection problems is the modularity Q defined in [45] as

$$Q=\sum_{c\in C}\left[\frac{m_c}{m}-\left(\frac{2m_c+e_c}{2m}\right)^2\right] \tag{7}$$

Where *m* is the number of links in the network, *mc* is the number of links in the community *c* and *ec* is the number of external links in the community *c*.

### 5.1.4 Density

In IsoFdp [44], an improved partition density function to evaluate the quality of the detected communities is proposed as follow:

$$D = \frac{2}{\sqrt{k}N}\sum_{c} n_c \frac{m_c-(n_c-1)}{(n_c-2)(n_c-1)} \qquad (8)$$

Where, N is the number of nodes in the network. $m_c$, $n_c$ are the numbers of links and nodes respectively, in each community $c$. $\sqrt{k}$ is the squared root of the community number.

## 5.2 Dataset 1

In order to generate the dataset we use the benchmark used in [11], which is like Girvan-Newman (GN) benchmark [46]. The network has 128 nodes, divided into four groups with 32 nodes each. The average degree of the network is noted $k$, and the number of edges connecting each node outside the community is noted $z$. To create a dynamic network we move $nC\%$ nodes from one community to another. The benchmark needs to parameter $N$ which represents the number of nodes in the network, and the parameters $cmin$, $cmax$ that illustrate respectively the maximum and the minimum community size. The benchmark also needs to the average degree $k^{-}$, the maximum degree $kmax$, the mixing parameter of the topology $\mu$, the number of steps $s$ to get a dynamic network, and the probability of switching $p$ which indicates how many nodes can change its community.

## 5.3 Dataset 2

As defined in [43] SYN-FIX is similar to the synthetic dataset 1. The network consists of 128 nodes divided into four communities of 32 nodes each. Every node has an average degree of 16 and shares a number z of links with the other nodes of the network. In order to introduce the dynamics, 3 nodes are randomly selected from each community and randomly assigned to the other three communities. Two datasets of type SYN-FIX are generated by varying the parameter z to 3 and 5, respectively.

## 5.4 Dataset 3

The SYN-VAR is a synthetic network composed of 256 nodes that are divided into four communities. As defined in [26], the benchmark includes the forming and dissolving process of communities and the attaching and detaching of nodes. At each time step, 32 nodes are randomly selected to form a new community. This is done for 5 time steps and these selected nodes return to their original communities in the following 5 time steps. In addition, 16 nodes are randomly deleted and 16 new nodes are added to network at each time step.

## 5.5 Real networks

**Cell phone calls** It is the dataset 1 from IEEE VAST 2008 Mini challenge 3, which includes 400 nodes and more than 498 edges.

**Karate** It is a social network of friendships between 34 members of a karate club at a US university in the 1970s. The network has 34 nodes and 78 edges.

**Football** This is the network of American football games between Division IA colleges during regular season Fall 2000. The network has 115 nodes and 613 edges.

**Dolphins** It is an undirected social network of frequent associations between 62 dolphins in a community living of Doubtful Sound, New Zealand. The network has 62 nodes and 159 edges.

**Lesmis** This undirected network contains co-occurrences of characters in Victor Hugo's novel 'Les Misérables'. A node represents a character, and an edge between two nodes shows that these two characters appeared in the same chapter of the book. The network has 77 nodes and 254 edges.

**Jazz** This is the collaboration network between Jazz musicians. Each node is a Jazz musician and an edge denotes that two musicians have played together in a band. The network has 198 nodes and 2742 edges.

## 5.6 Results in terms of NMI on dataset 1

In this section we test the method on two networks generated by varying the parameter $nC$ from 10% to 30%, setting the parameter $\overline{k}$ to 16 and the parameter $z = 5$ . The other parameters of the dataset are set as follow: $N = 128$, $s = 5$, $kmax = 17$, $cmin = 32$, $cmax = 32$. The obtained results are compared to the results of L-DMGA [11], DYNMOGA [43], FacetNet [8]. We refer to our method by MS-DYN.

From Figure Figs. 4 and 5 we vary the parameter $nC$ to get two different type of networks, on which ones we can see that MS-DYN conducts better than L-DMGA, DYNMOGA and FacetNet in terms of normalized mutual information. This means that our method approximates the ground truth on different type of networks and on different steps of the dynamic networks. In Fig. 4 at time $t = 20$ we can see that MS-DYN has 0.33 as value of NMI which is high than LDMGA,

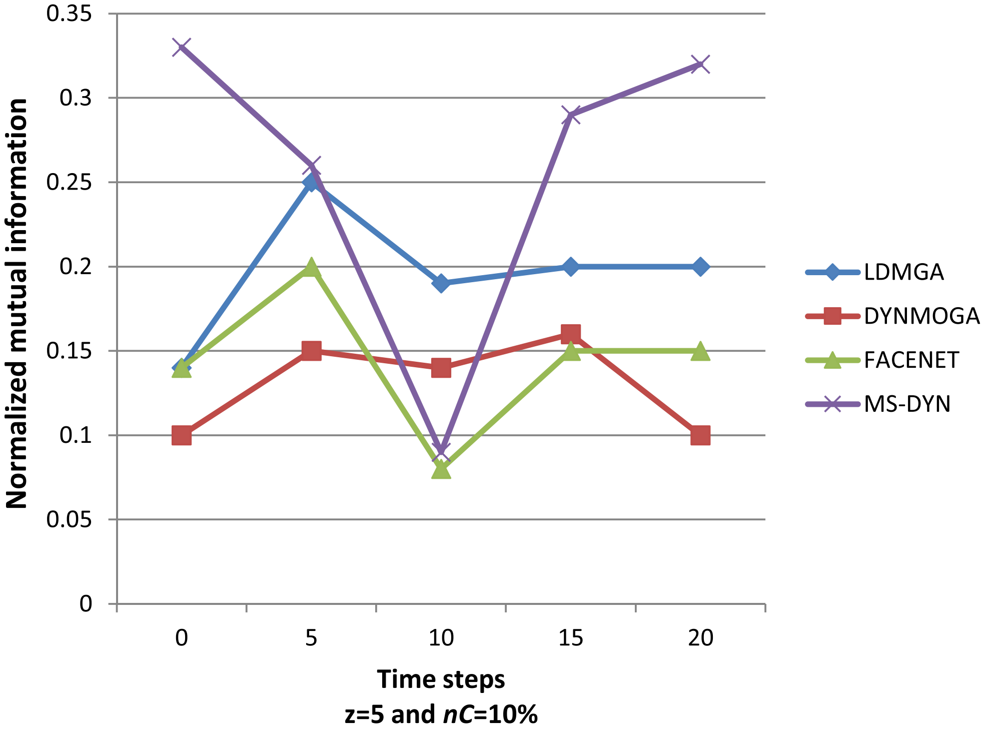


**Fig. 4** Normalized mutual information of detected communities where nC =10%

DYNMOGA and FacetNet which have respectively 0.2, 0.15 and 0.1 as values of NMI. In Fig. 5 at time $t = 10$ we can see that MS-DYN has 0.6 as value of NMI which is high than L-DMGA, DYNMOGA and FacetNet which have respectively 0.25, 0.15 and 0.18 as values of NMI.

### 5.7 Results in terms of NMI on dataset 2

In this section we test the method on Syn-fix network, where we set the parameter $z$ to 3 and 5. The obtained results are compared to the results of DYNMOGA [43], FacetNet [8] and DYN-MODPSO [26].

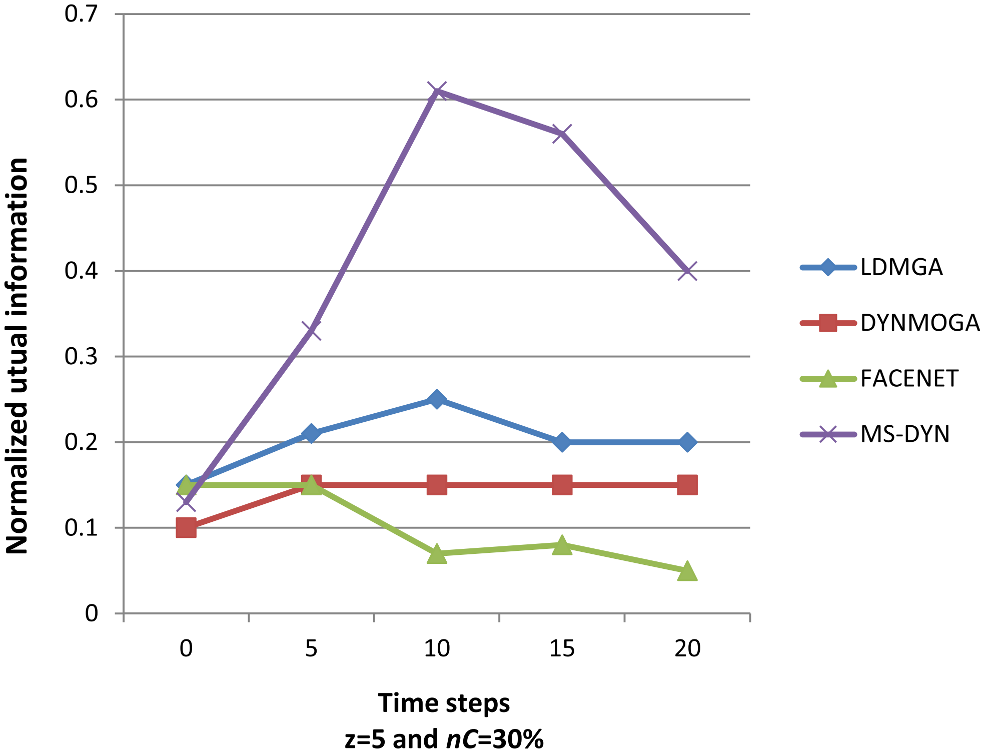
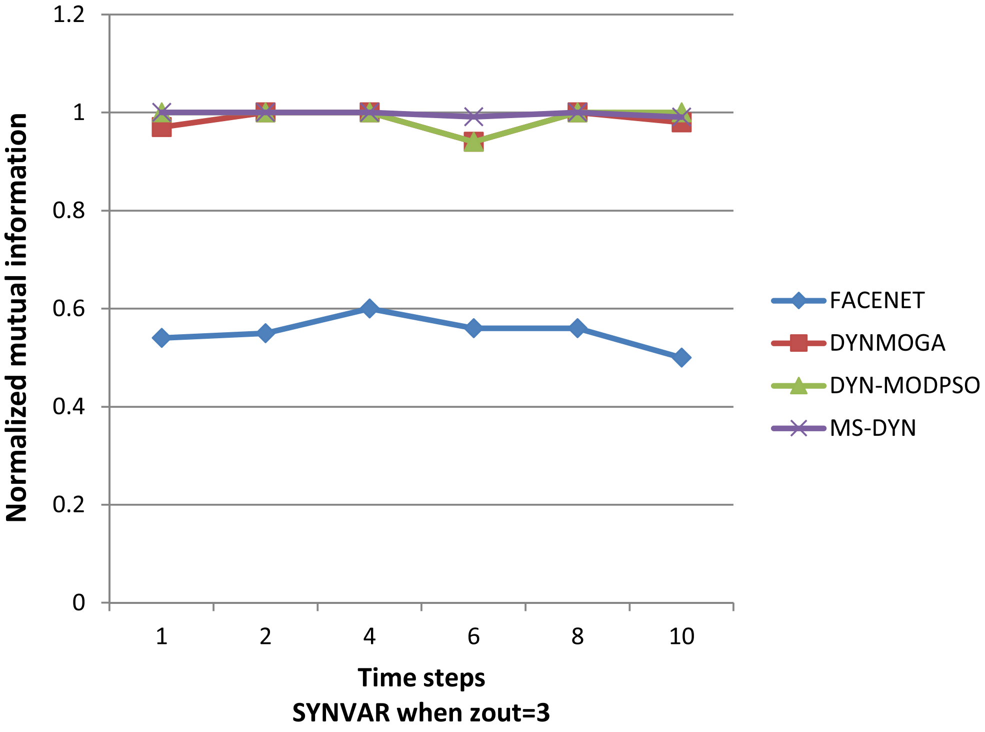


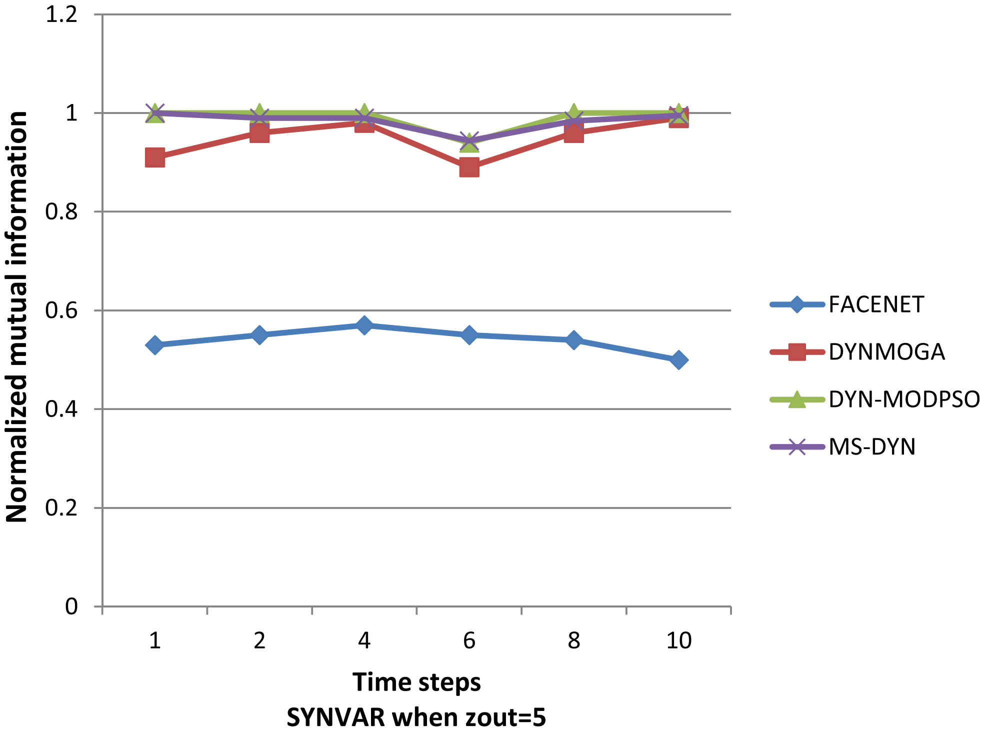
**Fig. 5** Normalized mutual information of detected communities where nC =30%

**Fig. 6** Normalized mutual information of detected communities on synfix network where z = 3



From Figs. 6 and 7 we can see that MS-DYN conducts better than DYNMOGA and FacetNet in terms of NMI and Error. For example, in Fig. 6 when FacetNet has 0.5 as value of NMI on all time steps, MS-DYN keeps the value of 1.0, which means that MS-DYN approximates well the ground truth. However MS-DYN keeps the same values of NMI as DYN-MODPSO on all time steps. In Fig. 7 in time *t* = 10, we can see that DYNMOGA fails to detect the real communities with value of NMI = 0.83, but MS-DYN still has the value of 1.0 as NMI.

**Fig. 7** Normalized mutual information of detected communities on synfix network where z = 5



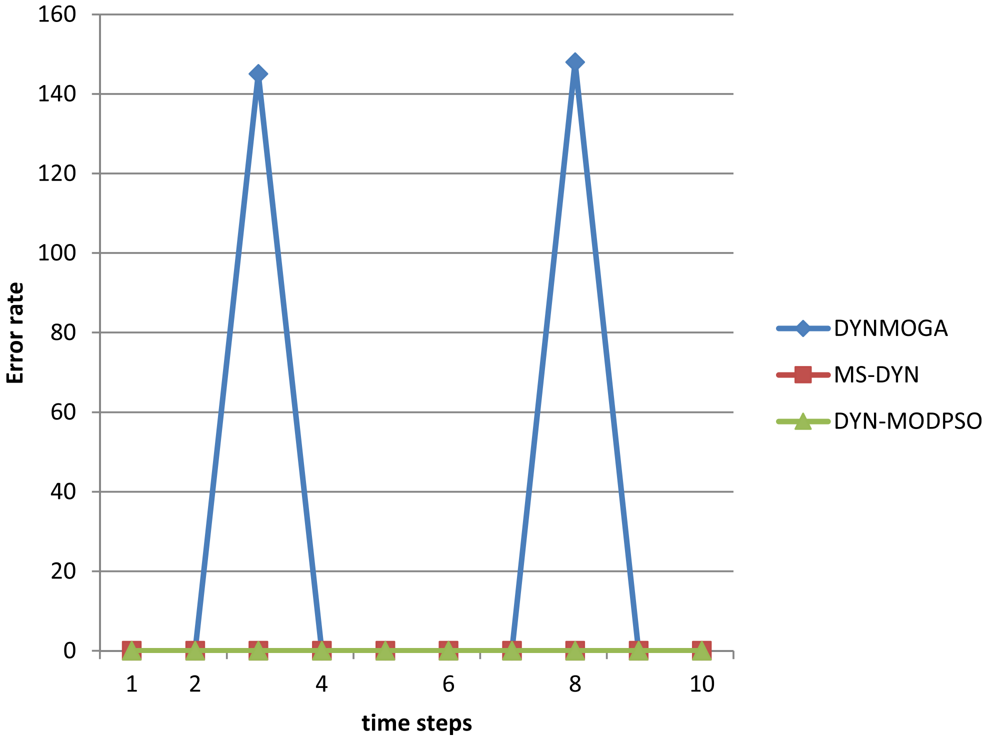


**Fig. 8** Error rate of detected communities on synfix network where z = 3

### 5.8 Results in terms of error rate on dataset 2

From Figs. 8 and 9 we can see that DYNMOGA has higher values of Error rate when z = 3 and when z = 5, compared to DYN-MODPSO and MS-DYN, which keep the value of 0 on all time steps of the two dynamic networks.

### 5.9 Results in terms of NMI on dataset 3

In this section we test the method on Syn-var network, where we set the parameter *z* to 3 and 5. The obtained results are compared to the results of DYNMOGA [43], FacetNet [8] and DYN-MODPSO [26].

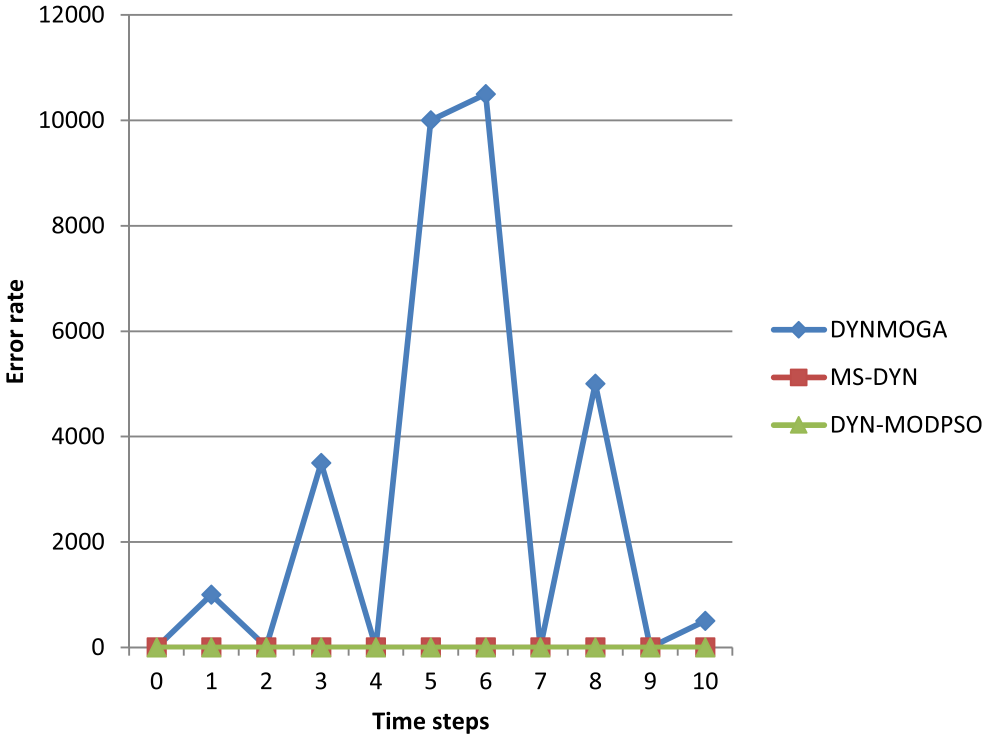


**Fig. 9** Error rate of detected communities on synfix network where z = 5

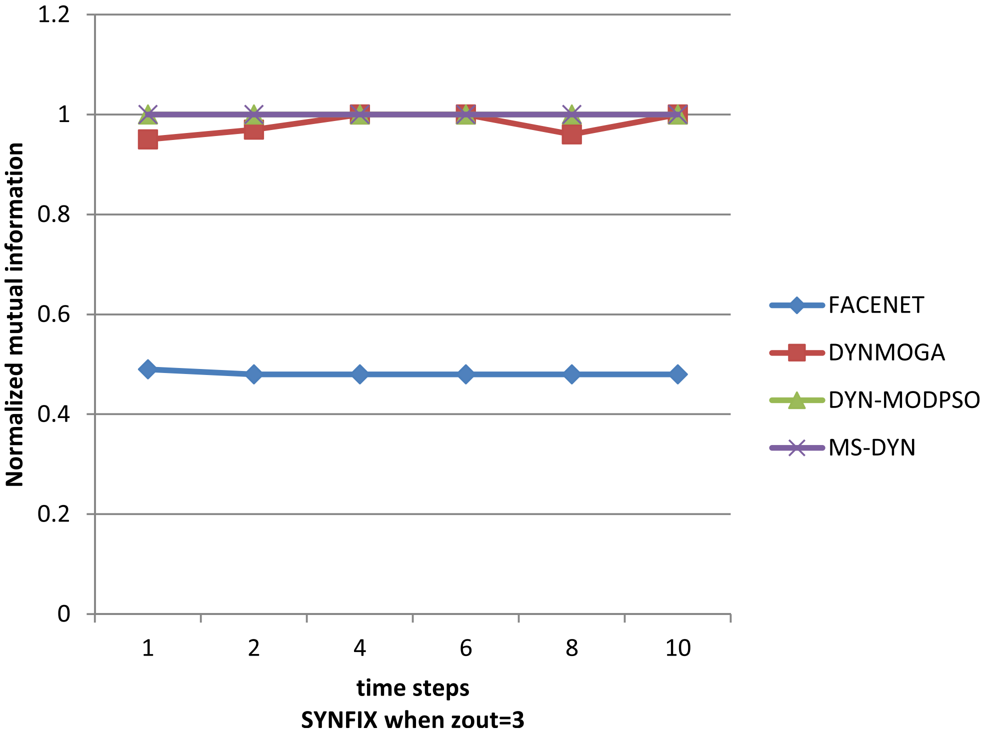


**Fig. 10** NMI of detected communities on Syn-var network where z = 3

In Figs. 10 and 11 we can see that FaceNet still has lower values of NMI which are between 0.5 and 0.6. However DYNMOGA has improved values of NMI which are between 0.9 and 1. In contrast, MS-DYN and DYN-MODPSO share the same values of NMI on most time steps, when z = 3 and z = 5. For example, when z = 3 and *t* = 6, DYN-MODPSO has lower value of NMI compared to MS-DYN, furthermore when z = 5 and t = 6, DYN-MODPSO and MS-DYN have NMI less than 1 but MS-DYN still outperforms DYN-MODPSO.

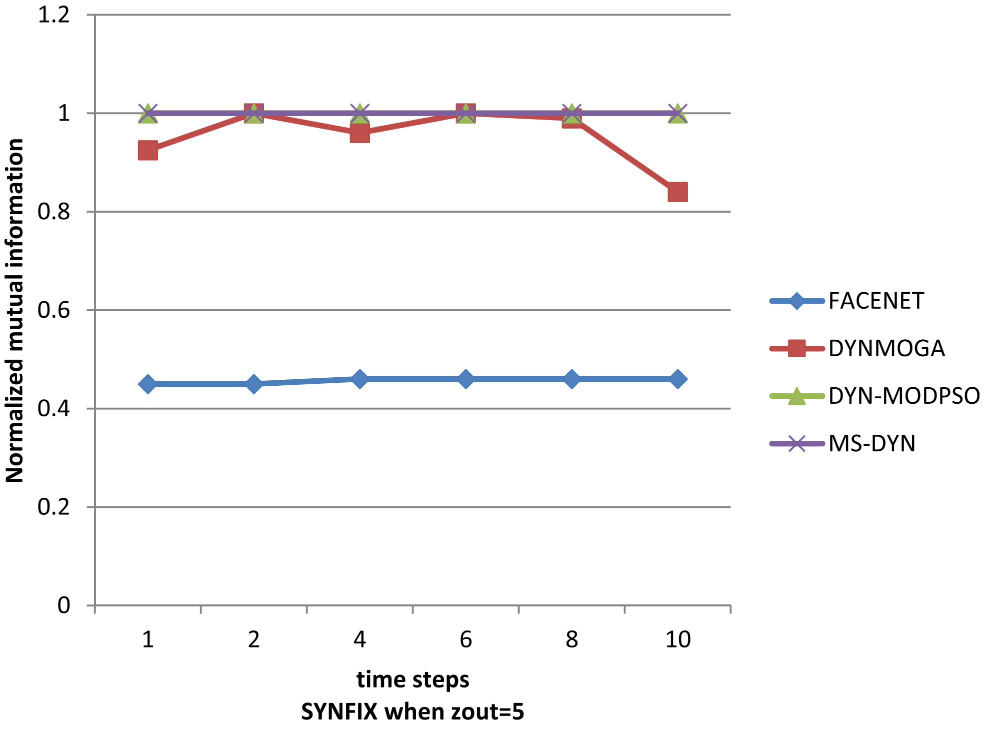


**Fig. 11** Normalized mutual information of detected communities on Syn-var where z = 5

### 5.10 Results in terms of modularity on cell phone calls network

In this section, we compared the obtained results in terms of Modularity to the results of DYNMOGA [43], DYN-MODPSO [26] and ESPRA [24].

In Fig. 12 we can see that ESPRA has lower values of modularity over all time steps compared to DYNMOGA and DYN-MODPSO which have approximated values. However MS-DYN outperforms twice DYNMOGA and DYNMODPSO on the most time steps.

### 5.11 Results in terms of modularity on real networks

In this section, we compared the obtained results in terms of Modularity to the results of MOGA-Net [47], CNLPSO-DE [48] and MOEA [49].

From Table 2 we can see that MS-DYN outperforms the other evolutionary algorithms on most real networks.

**Table 2** Results in terms of Modularity on Real network

| | Karate | Dolphins | Football | Net-science |
|---|---|---|---|---|
| MS-DYN | 0.6346 | 0.8843 | 0.8112 | 0.8969 |
| MOGA-Net | 0.4198 | 0.5258 | 0.5280 | 0.8916 |
| CNLPSO-DE | 0.420 | 0.5265 | 0.6045 | 0.9519 |
| MOEA | 0.4198 | 0.5210 | 0.6044 | 0.9143 |

### 5.12 Results in terms of density on real networks

In this section, we compared the obtained results to those of IsoFdp [44] in terms of Density.

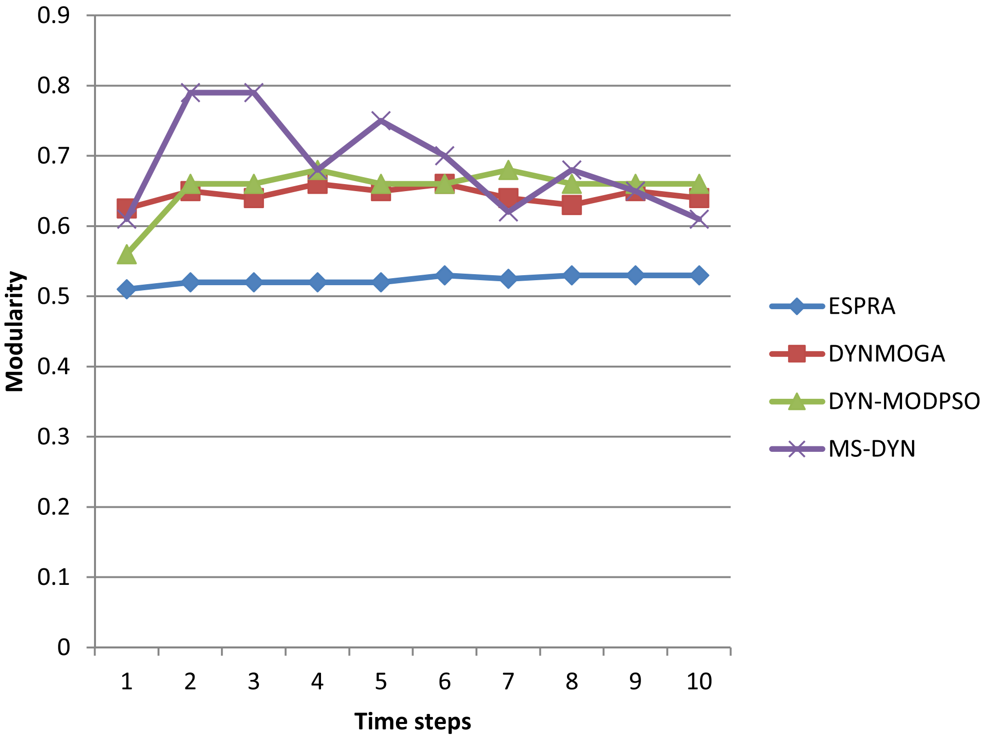
From Fig. 13 we can see that MS-DYN outperforms IsoFdp in terms of density, on dolphins and jazz networks. The IsoFdp outperforms MS-DYN on football and jazz networks. However, on football network, MS-DYN has 5 communities compared to IsoFdp which has 8 communities. On lesmis, also IsoFdp outperforms MS-DYN, but MS-DYN has 4 communities which are lower than 5 communities of IsoFdp.

In Fig. 14 we show the partition of SYN-FIX when z = 3 by MS-DYN algorithm.

### 5.13 Discussion

This paper presents a multi-objective based community detection method. The method uses MS approach to generate the individuals of high quality; these individuals are optimized using a multi- objective BA. After getting the best one, we decode it to have the corresponding communities. By using two objective functions we get a set of best solutions rather than one. In addition, to generate a new solution we use the velocity and

**Fig. 12** Modularity of detected communities on Cell Phone Calls network



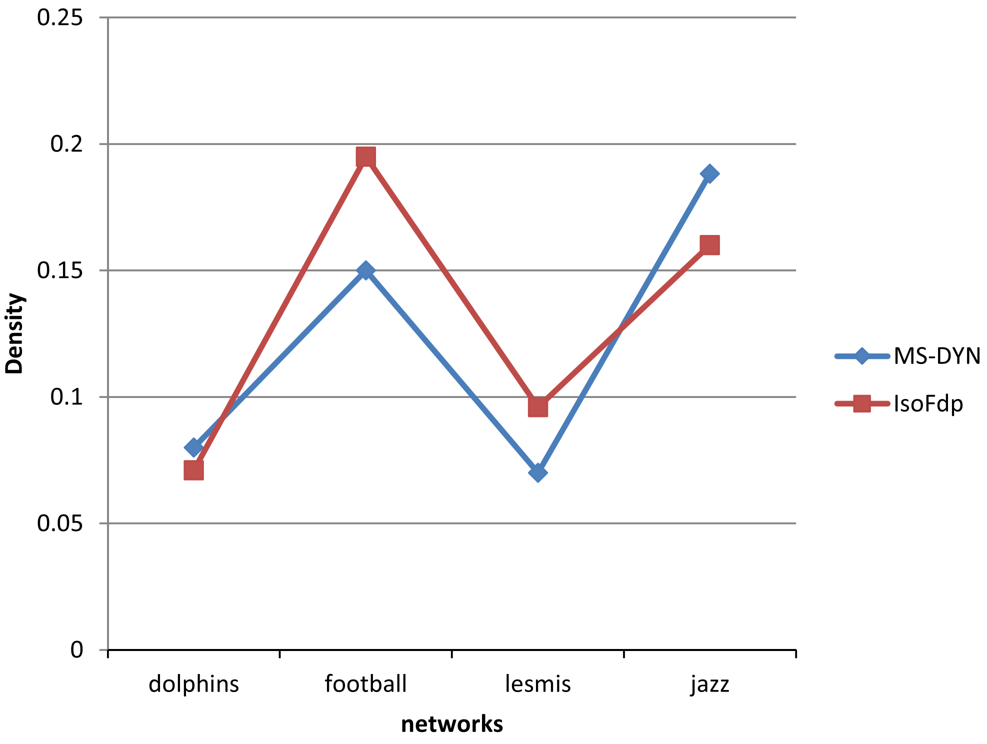


**Fig. 13** Density of detected communities on real networks

the frequency of BA which returns the solutions of good quality. Another advantage of multi-objective BA is the assignment of weight to objective functions which facilitates the non-dominated solution search. This assignment is better than the fast non-dominated sort of genetic algorithm that uses the crowding distance.

## 6 Conclusion

Many algorithms have been proposed to deal with the problem of community detection in static networks, but the evolution of the networks over the time implies a lot of changes which motivates us to treat the community detection problem on

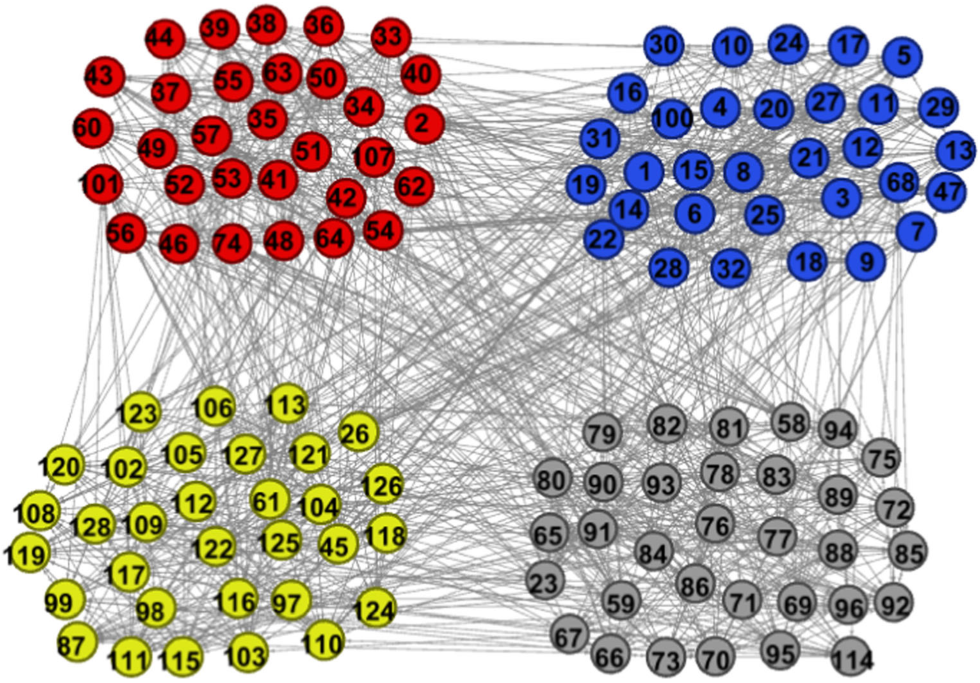


**Fig. 14** NMI of detected communities on SYN-FIX where z = 3

dynamic network. The major community detection evolutionary methods generate the initial population randomly which need a lot of iterations to get good solutions. In this paper, we use MS approach to generate high quality population. MS does not need to fix the number of clusters in advance as k-means. The used BA also does not need to any parameters; the algorithm simulates the movement of real bats to artificial bats by giving another signification to the velocity and frequency. The individuals in the algorithm are evaluated using the modularity density and NMI as objective functions. MS-DYN showed better performance than the compared algorithms in terms of NMI and *SQt* testing on artificial dynamic networks. MS-DYN also showed better performance than the compared algorithms in terms of modularity and density on real dynamic networks. The performance of our method is derived from the convergence of BA and the high quality results of MS approach. In future work, we plan to track the evolution of communities over the time. Since, one community can be merged with another community, split or died...etc. So we can consider this problem as classification problem and take all this events as classes.

## References

1. Xie, J., B. K. Szymanski, and X. Liu (2011) Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In 2011 ieee 11th international conference on data mining workshops pp. 344–349
2. Bello-Orgaz G, Camacho D (2014) Evolutionary clustering algorithm for community detection using graph-based information. In: 2014 IEEE Congress on Evolutionary Computation Beijing, CEC 2014, pp 930–937
3. Belkhiri Y, Kamel N, Drias H, Yahiaoui S (2017) Bee Swarm Optimization for Community Detection in Complex Network. In: Rocha Á, Correia A, Adeli H, Reis L, Costanzo S (eds) Recent Advances in Information Systems and Technologies. WorldCIST 2017. Advances in intelligent systems and computing, vol 570. Springer, Cham
4. Imane M, Kamel N (2018) Overlapping community detection with a novel hybrid metaheuristic optimization algorithm. International Journal of Data Mining, Modelling and Management
5. Duan D, Li Y, Li R, Lu Z (2012) Incremental K-clique clustering in dynamic social networks. Artif Intell Rev 38:129–147
6. Enugala R, Rajamani L, Ali K, Kurapati S (2015) Community detection in dynamic social networks: a survey. International Journal of Research and Applications 2(6):278–285
7. Liu Q et al (2016) Evolutionary link community structure discovery in dynamic weighted networks. Physica A
8. Lin Y-R, Chi Y, Zhu S et al (2008) A framework for analyzing communities and their evolutions in dynamic networks. In: proceedings of the 17th international conference on world wide web, pp 685–694
9. Folino F, Pizzuti C (2014) An evolutionary multiobjective approach for community discovery in dynamic networks. IEEE Trans Knowl Data Eng 26:1838–1852
10. Zhou X, Liu Y, Li B, Sun G (2015) Multiobjective biogeography based optimization algorithm with decomposition for community detection in dynamic networks. Physica A: Statistical Mechanics and its Applications 436:430–442
11. Niu X, Si W, Wu CQ (2017) A label-based evolutionary computing approach to dynamic community detection. Comput Commun 108: 110–122
12. Zhang X-K, Ren J, Song C, Jia J, Zhang Q (2017) Label propagation algorithm for community detection based on node importance and label influence. In: Physics Letters A, vol 381, No. 33, pp 2691–2698
13. Newman MEJ (2004) Fast algorithm for detecting community structure in networks. Phys Rev E 69(6):066133+. https://doi.org/10.1103/physreve.69.066133
14. V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre (2008) Fast unfolding of communities in large networks. J Stat Mech: Theory Exp, 2008 (10), p. P10008
15. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci 105(4):1118–1123
16. Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106
17. Pizzuti C (2012) A multiobjective genetic algorithm to find communities in complex networks. Evol Comput IEEE Trans 16(3): 418–430
18. Ji J, Song X, Liu C, Zhang X (2013) Ant colony clustering with fitness perception and pheromone diffusion for community detection in complex networks. Physica A: Statistical Mechanics and its Applications 392(15):3260–3272
19. Lili FU et al (2014) Research on Spectral Clustering. Appl Mech Mater 687–691:1350–1353
20. Hu Y, Yang B, Lv C (2016) A local dynamic method for tracking communities and their evolution in dynamic networks. Knowl-Based Syst 110:176–190
21. Chang-Dong W, Jian-Huang L, Philip SY (2014) NEIWalk: Community Discovery in Dynamic Content Based Networks. IEEE Trans Knowl Data Eng 26 No. 7
22. Nagehan I, Sule GO (2013) Community Event Prediction in Dynamic Social Networks. In: proceedings 12th IEEE International Conference on Machine Learning and Applications, pp 191–196
23. Asur S, Parthasarathy S, Ucar D (2009) An event-based framework for characterizing the evolutionary behavior of interaction graphs. Trans Knowl Discov Data, ACM 3:16
24. Peizhuo Wang et al (2017). Dynamic community detection based on network structural perturbation and topological similarity. J Stat Mech Theory Exp 2017. https://doi.org/10.1088/1742-5468/2017/1/013401
25. Attea B.A (2016) A new multi-objective evolutionary framework for community mining in dynamic social networks. Swarm and Evolutionary
26. Gao C et al (2018) Multiobjective discrete particle swarm optimization for community detection in dynamic networks. EPL 122(2): 28001
27. Augusto O, Fouad B, Caro S (2012) A new method for decision making in multi-objective optimization problems. Pesqui Oper 32(2):331–369
28. Yammani C, Maheswarapu S, Matam SK (2016) A multi-objective shuffled bat algorithm for optimal placement and sizing of multi distributed generations with different load models. Int J Electr Power Energy Syst 79:120–131
29. Tharakeshwar TK, Seetharamu KN, Durga Prasad B (2017) Multi-objective optimization using bat algorithm for shell and tube heat exchangers. Appl Therm Eng 110:1029–1038
30. Yang N-C, Le M-D (2015) Optimal design of passive power filters based on multi-objective bat algorithm and pareto front. Appl Soft Comput 35:257–266 ISSN 1568-4946

31. Liang H, Liu Y, Li F, Shen Y (2018) A multiobjective hybrid bat algorithm for combined economic/emission dispatch. Int J Electr Power Energy Syst 101:103–115 ISSN 0142-0615

32. Yang XS (2010) A New Metaheuristic Bat-Inspired Algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N (eds) Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence, vol 284. Springer, Berlin, Heidelberg

33. Laamari MA, Kamel N (2014) A hybrid bat based feature selection approach for intrusion detection. In: Pan L, Păun G, Pérez-Jiménez MJ, Song T (eds) Bio-Inspired Computing - Theories and Applications. Communications in Computer and Information Science, vol 472. Springer, Berlin, Heidelberg

34. Heraguemi KE, Kamel N, Drias H (2014) Association rule mining based on bat algorithm. In: Pan L, Păun G, Pérez-Jiménez MJ, Song T (eds) Bio-Inspired Computing - Theories and Applications. Communications in Computer and Information Science, vol 472. Springer, Berlin, Heidelberg

35. Heraguemi K, Kamel N, Drias H (2018) Multi-objective bat algorithm for mining numerical association rules. International Journal of Bio-Inspired Computation 11(239). https://doi.org/10.1504/IJBIC.2018.10013987

36. Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 26:369–395

37. Yang XS (2011) Bat algorithm for multiobjective optimization. Int J Bio-Inspired Computation 3(5):267–274

38. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5): 603–619

39. Shang R et al (2016) A multiobjective evolutionary algorithm to find community structures based on affinity propagation. Physica A 453:203–227

40. Hu Y-Q, Li M-H et al (2008) Community detection by signaling on complex networks. Phys Rev E 78

41. Li Z, Zhang S, Wang RS, Zhang XS, Chen L (2008) Quantitative function for community detection. Phys Rev E 77(3):036–109

42. MacKay DJC (2002) Information theory, inference and learning algorithms. Cambridge University Press, Cambridge

43. Folino F, Pizzuti C (2014) An evolutionary multiobjective approach for community discovery in dynamic networks. IEEE Trans Knowl Data Eng Vol. 99, No. 8

44. You T, Cheng H-M, Ning Y-Z, Shia B-C, Zhang Z-Y (2016) Community detection in complex networks using density-based clustering algorithm and manifold learning. Physica A: Statistical Mechanics and Its Applications 464:221–230

45. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69

46. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E Stat Nonlinear Soft Matter Phys 78:046110. https://doi.org/10.1103/PhysRevE.78.046110

47. Pizzuti C (2012) A multiobjective genetic algorithm to find communities in complex networks. IEEE Trans Evol Comput 16(3): 418–430

48. Pourkazemi M, Keyvanpour MR (2017) Community detection in social network by using a multi-objective evolutionary algorithm. Intelligent Data Analysis 21:385–409

49. Gong M, Ma L, Zhang Q, Jiao L (2012) Community detection in networks by using multiobjective evolutionary algorithm with decomposition. Physica A, Elsevier 391:4050–4060