Knowledge-Based Systems 46 (2013) 1–11

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Community Detection in Complex Networks: Multi–objective Enhanced Firefly Algorithm

Babak Amiri [a,*], Liaquat Hossain [a], John W. Crawford [b], Rolf T. Wigand [c]

[a] Center for Complex Systems Research, The University of Sydney, Australia
[b] Charles Perkins Centre, The University of Sydney, Australia
[c] Departments of Information Science and Management, University of Arkansas at Little Rock, United States

ARTICLE INFO

*Article history:*
Received 16 March 2012
Received in revised form 23 November 2012
Accepted 5 January 2013
Available online 14 February 2013

*Keywords:*
Complex network
Community
Multi-objective
Enhanced firefly algorithm
Pareto-optimal front

ABSTRACT

Studying the evolutionary community structure in complex networks is crucial for uncovering the links between structures and functions of a given community. Most contemporary community detection algorithms employs single optimization criteria (i.e.., modularity), which may not be adequate to represent the structures in complex networks. We suggest community detection process as a Multi-objective Optimization Problem (MOP) for investigating the community structures in complex networks. To overcome the limitations of the community detection problem, we propose a new multi-objective optimization algorithm based on enhanced firefly algorithm so that a set of non-dominated (Pareto-optimal) solutions can be achieved. In our proposed algorithm, a new tuning parameter based on a chaotic mechanism and novel self-adaptive probabilistic mutation strategies are used to improve the overall performance of the algorithm. The experimental results on synthetic and real world complex networks suggest that the multi-objective community detection algorithm provides useful paradigm for discovering overlapping community structures robustly.



## 1. Introduction

The modern science of networks is probably the most active field within the new interdisciplinary science of complex systems. Many complex systems can be represented as networks, where the elementary parts of a system and their mutual interactions are nodes and links, respectively. Finding compartments may shed light on the organization of complex systems and on their function [1–4]. Therefore, detecting communities in networks has become a fundamental problem in network science. Many methods have been developed, using tools and techniques from disciplines like physics, biology, applied mathematics, computer and social sciences.

A recent analysis of community detection and network structure has attracted the attention of researches in deferent areas. It can be considered almost like an optimization problem [5] that is very difficult to solve, so lots of studies have been done based on evolutionary methods like GA [6–11], SA [12] and collaborative evolutionary algorithms [13,14].

In order to find good communities we need a proper quality function that should be intuitive and easy to agree upon. To evaluate the goodness of a community having a quantitative criterion will be helpful and sometimes even necessary. A quality function assigns a number to each community of networks, and communities can be ranked based on the scores of these quality functions. Communities with high scores are good, but determining when one cluster is better than another depends on the specific concept of community and/or quality function adopted. There is no one universally accepted definition of a quality function because it depends on the specific system at hand and/or its application. There is a reference guideline at the basis of most community definitions, but there must be more links inside the community than links connecting nodes of the community with the rest of the network. In this way, detecting problems in the community can be formulated with two different objectives, maximization of internal links and minimization of external links [15]. The relationship between these two objectives is a trade off because they are correlated, so community detection can be formulated as a multi-objective optimization problem to which this study proposed a multi-objective algorithm based on enhanced firefly algorithm.

The original Firefly Algorithm (FA) is a new population based evolutionary algorithm inspired by the brightness and attractiveness of fireflies seen in the summer sky in some regions, which they use to protect themselves from predators and absorb their prey [16,17]. In an optimization problem, fireflies move towards brighter points to find a global optimum solution. However, the performance of the original FA greatly depends on its parameters

* Corresponding author.
*E-mail address:* amiri.babak@sydney.edu.au (B. Amiri).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.01.004

such as its attractive coefficients and random movement factor, even though it suffers from the problem of being trapped in its own local optima. In this paper, a chaotic sequence mechanism was used to tune the random movement factor based on [18] and the absorption parameter was set to one according to [16]. Also, a novel self adaptive probabilistic mutation strategy is highly recommended to enhance the algorithm's performance. The augmented mutation strategy was implemented to improve the convergence characteristic and the quality of the solution. During the simulation, a set of non-dominated solutions were stored in an external memory (repository) whose size was controlled by a fuzzy-based clustering technique. A niching mechanism was used to select the best compromise solution from the repository such that the population would move towards a smaller search space in the Pareto-optimal front. A min–max approach was also used to select the best candidates for the next iteration in order to involve the decision maker's favor through the entire search process. Here, we propose a novel approach to select the population of the next iteration, which enables the algorithm to obtain a uniform POF and include the extreme points of the surface trade-off.

The proposed algorithm optimized two objective functions, the community score that measures the density of the clusters obtained and community fitness that minimizes the external links. A prior knowledge of a number of communities was not needed because this method returns a set of solutions where each of them correspond to different trade-offs between the two objectives, and gives a great chance to analyze the hierarchy of communities.

The rest of the paper is organized as follows. Section 2 introduces the problem of community detection. The concept of a multi-objective optimization problem is reviewed in Section 3. The original firefly algorithm is explained in Section 4. In Section 5, the proposed enhanced firefly algorithm (CICA) to detect community is presented, and then in Section 6, the experimental results of the proposed algorithm in comparison with other approaches are shown.

## 2. Community detection

### 2.1. Literature review

Community detection in complex networks has been studied in multiple fields for years, particularly computer science and physics. Traditional graph partitioning methods, such as Kernighan–Lin algorithm [19], Girvan–Newman algorithm [20], normalized cut [21], and spectral bi-section methods [22] have been used widely to find network communities. Recently, significant progress has been archived in this research field and many approaches for detecting communities in networks have been presented.

Modularity based methods: To evaluate the quality of network partitions, Newman and Girvan proposed a modularity measure $Q$ [23], which has been widely used in community discovery. Modularity-based methods assume that high values of modularity indicate good partitions, but it has been proven that optimizing modularity is a complete NP problem. Most of the modularity based algorithms find good approximation of maximum modularity with high computational complexity such as SA (Simulated Annealing) [24], FN [25,19], and CNM [26,6]. Recently, Blondel et al. proposed a greedy modularity based algorithm called BGLL [27,3], for finding communities in weighted networks. This algorithm has a low computational complexity and can discover hierarchical communities, but the results depend on the order in which the nodes are visited. Actually, the methods of greedy optimization of modularity often tend to form large communities through a combination of small ones. Recent research shows that modularity is not a scale-invariant measure, and hence, by relying on its maximization, detecting communities smaller than a certain size is impossible. This serious problem is famously known as the resolution limit of modularity based algorithms [28,10]. Compared to the traditional modularity based methods, this paper used modularity as a quality function to guide the selection of optimal communities.

Hierarchical and Overlapping methods: In the presence of hierarchy, the concept of community structure becomes richer. Agglomerative or divisive hierarchical clustering is a well known technique used to solve this problem [20,25]. Starting from a partition where each node is its own community, or where all the nodes are in the same community, one merges or splits clusters according to a topological measure of similarity between the nodes. In this way, one builds a hierarchical tree of partitions. Although this method naturally produces a hierarchy of partitions, it needs a metric to stop the algorithm. Some recent work focused on the problem of identifying meaningful community hierarchies [29] and detecting multi-resolution levels [30–32].

The issue of finding overlapping communities has become an important topic. Palla et al. proposed a clique percolation method (CPM) [33]. A complete sub-graph of $k$ nodes, called $k$-clique, is rolled over the network through other cliques with $k-1$ common nodes. In this way, a set of nodes regarded as a community, can be reached. One node can belong to more than one community, therefore, overlaps naturally occur. The CPM algorithm is limited by its assumption that a graph has a large number of cliques. Furthermore, this is not a suitable method for detecting the hierarchical structure. Nepusz et al. recently considered the problem of fuzzy community detection in networks, which expands the concept of overlapping the community structure [34], and where every node is allowed to belong to multiple communities with different degrees of membership. A measure was introduced to identify regular nodes in a community such as hubs that have a significant membership in more than one single community, and outliers that do not belong to any of the communities.

In real networks, communities are both usually hierarchical and overlapping. Most existing methods investigate these two phenomena separately. Our work is one of the few methods that try to discover both hierarchical communities and overlapping nodes in a given network.

*Density based methods*: Density based clustering approaches (e.g.. DBSCAN [35] and OPTICS [36]) have been widely used in data mining owing to their ability of finding clusters with an arbitrary shape, even in the presence of noise. Recently, Xu et al. proposed an efficient structural network clustering algorithm SCAN [37] by extending the DBSCAN [35]. This algorithm can find communities as well as hubs and outliers in a network but it does require a minimum similarity parameter $\varepsilon$ and a minimum size cluster $\mu$ to define clusters, and is sensitive to the parameter $\varepsilon$, which is difficult to determine automatically. To deal with this problem Bortner et al. proposed a new algorithm, called the SCOT + HintClus [38], to detect the hierarchical cluster boundaries of networks by extending the algorithm OPTICS [36]. However, it cannot find the result of global clustering because it needs an additional pruning process to expose the reasonable hierarchical structure of the networks. Our work tries to develop a method free from parameters to explore the hierarchy of structurally connected communities with multi-resolution levels in the networks.

### 2.2. Community detection problem

In this study, an undirected network $G = (V, E)$ defined by a set of nodes $\{V\}$ or vertices, and a set of links $\{E\}$ connect two elements of $V$. A community consists of vertices and an edge between these nodes, where the nodes often cluster into tightly knit groups with a high density and a lower density of between the group connections [39].

A network can be represented mathematically by an adjacency matrix $A$, if there is an edge from $v_i$ to $v_j$, $A_{ij} = 1$ and $A_{ij} = 0$ otherwise. The degree $k$ i of a node $i$, defined as $k_i = \sum_j A_{ij}$. Let $C \subset G$ the sub-graph where node $i$ belongs to, the degree of $i$ with respect to $C$ can be split as $k_i(C) = k_i^{in}(C) + k_i^{out}(C)$, where $k_i^{in}(C) = \sum_{j \in C} A_{ij}$ is the number of edges connect the $i$ to the other nodes in $C$, and $k_i^{out}(C) = \sum_{j \notin C} A_{ij}$ is the number of edges connecting $i$ to the rest of the network. A sub-graph $C$ is a community in a strong sense if $k_i^{in}(C) > k_i^{out}(C)$, $\forall i \in C$. A sub-graph $C$ is a community if $\sum_{i \in C} k_i^{in}(C) > \sum_{i \in C} k_i^{out}(C)$.

The quality measure of a community $C$ that maximizes the in-degree of the nodes belonging to $C$ has been introduced in [6]. On the other hand, in [17] a criterion that minimizes the out-degree of a community is defined. We now recall the definitions of these measures first, and then we show how they can be exploited in a multi-objective approach to find communities. In the following, without losing its generality, the network graph is assumed to be undirected.

Let $\mu_i$ denote the fraction of edges connecting node $i$ to the other nodes in $C$. More formally, $\mu_i = \frac{1}{|C|} k_i^{in}(C)$ where $|C|$ is the cardinality of $C$.

The power mean of C of order r, denoted as $M(C)$ is defined as

$$M(C) = \frac{\sum_{i \in C} (\mu_i)^r}{|C|} \tag{1}$$

Notice that in the calculation of $M(C)$, since $0 \leqslant \mu \leqslant 1$, the exponent $r$ increases the weight of nodes having many connections with other nodes belonging to the same module, and diminishes the weight of those nodes having few connections inside $C$.

The volume $v_C$ of a community $C$ is defined as the number of edges connecting vertices inside $C$, i.e. the number of 1 entries in the adjacency sub-matrix of $A$ corresponding to $C$, $v_C = \sum_{i,j \in C} A_{ij}$.

The score of $C$ is defined as $score(C) = M(C) \times v_C$. Thus, the score takes into account both the fraction of interconnections among the nodes (through the power mean) and the number of interconnections contained in the module $C$ (through the volume). The community score of a clustering $\{C_1, \ldots, C_k\}$ of a network is defined as

$$CS = \sum_{i=1}^{k} score(C_i) \tag{2}$$

The community score gives a global measure of the network division in communities by summing up the local score of each module found. The problem of community detection has been formulated in [9] as the problem of maximizing $CS$.

In [17] the concept of community fitness of a module $C$ is defined as

$$P(C) = \sum_{i \in S} \frac{k_i^{in}(C)}{\left(k_i^{in}(C) + k_i^{out}(C)\right)^{\propto}} \tag{3}$$

where $k_i^{in}(C)$ and $k_i^{out}(C)$ are the internal and external degrees of the nodes belonging to the community $C$, and $\alpha$ is a positive real-valued parameter controlling the size of the communities. The community fitness has been used by [31] to find communities.

## 3. Solution methodology

The proposed solution methodology comprises two steps. In the first step, the Pareto-based Multi-objective Optimization Problem (MOP) is described and then the novel Enhanced Firefly Algorithm (EFA) is used to extract the Pareto Optimal Front (POF).

### 3.1. Pareto-based approach for the community detection problem

Many real world optimization problem consists of more than one commensurable and conflicting objective functions, and thus there is no single optimal solution that simultaneously optimizes all the objective functions. Hence, in the absence of any preferred information, a non-dominated set of solutions were obtained [40,41]. In this paper, the community detection problem was solved by an efficient EFA to reach a set of Pareto-optimal solutions in a single run.

#### 3.1.1. Definition of MOP

The MOP can be modeled as the following equations:

$$\min \; F = [f_1(X), f_2(X), \ldots, f_n(X)] \tag{4}$$

where $f_i(X)$ is the $i$th objective function and $X$ is the vector of the optimization variables, $n$ is the number of objective functions.

The solution to the multi-objective optimization problem is a set of Pareto points. In the multi-objective optimization problem, a solution $X^* \in \Omega$ is a Pareto optimal if there is no solution $(X)$ in $\Omega$ such that $X$ dominates $X^*$. $\Omega$ is the set of all feasible values of $X$. The solution $X_1$ is said to dominate the solution $X_2$ if

$$\begin{aligned} &\forall j \in \{1, 2, \ldots, n\}, \quad f_j(X_1) \leqslant f_j(X_2) \\ &\exists k \in \{1, 2, \ldots, n\}, \quad f_k(X_1) \leqslant f_k(X_2) \end{aligned} \tag{5}$$

Solutions which dominate others but not themselves, are called non-dominated solutions.

#### 3.1.2. External repository

The Enhanced Firefly Algorithm (EFA) uses an external repository, which acts as an elite archive to store the non-dominated solutions. At the end of each iteration, after calculating two objective functions for each individual along the optimization process, the non-dominated procedure for each of the individuals was checked with the other individuals using (26) and (27). The non-dominated solutions selected were stored in the repository and the dominated members of the repository were deleted. It should be noted that the repository was initialized with the non-dominated solutions found in the initial population.

#### 3.1.3. Preserving diversity via the niching mechanism

The general aim of the MOP is not only to convey the solution search space to the POF, but also to preserve the diversity of the POF in its proper condition [40]. In this regard, it is very important to select the best global solution (as $\mathbf{Best}^k$) from the set of non-dominated solutions to prevent all the solutions from drifting in the same region. Thus, a niching fitness sharing technique was used to choose the $\mathbf{Best}^k$ from the non-dominated solutions in the repository. In this approach the sharing distance $d_{n_1 n_2}$ and sharing fitness function value $Sh(d_{n_1 n_2})$ were calculated as follows [40]:

$$d_{n_1 n_2} = \sqrt{\sum_{q=1}^{2} \left( \frac{F_q(X_{n_1}) - F_q(X_{n_2})}{F_q^{\max} - F_q^{\min}} \right)^2} \tag{6}$$

$$Sh(d_{n_1 n_2}) = \begin{cases} 1 - \left(\frac{d_{n_1 n_2}}{r_{share}}\right)^2 & \text{if } d_{n_1 n_2} \leqslant r_{share} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The $Sh(d_{n_1 n_2})$ returned the niche count $M_{n_1}$ in which the $n_1$th fitness function was proportional to the inverse of $M_{n_1}$. Then the $\mathbf{Best}^k$ was selected from the non-dominated solutions according to the Roulette Wheel Mechanism (RWM) [42]. The RWM improves the niching selection in its process. The niching approach can be expressed as follows:

Normalize the fitness function and use RWM to select the **Best**$^k$. The initial values for $M_{n_1}, n_1 = 1, \ldots, N_{rep}$ were zero.

The niching approach can be expressed as follows:

*For* $n_1$ = 1 to $N_{rep}$
  *For* $n_2$ = 1 to $N_{rep}$
    Calculate $d_{n_1 n_2}$ and $Sh(d_{n_1 n_2})$ according to (6) and (7).
    $M_{n_1} = M_{n_1} + Sh(d_{n_1 n_2})$
  *End For* (it refers to set $n_2$)
  $fit_{n_1} = \frac{1}{M_{n_1}}$
*End For* (it refers to set $n_1$)

Normalize the fitness function and apply RWM for selecting the **Best**$^k$. The initial values for $M_{n_1}, n_1 = 1, \ldots, N_{rep}$ are zero.

#### 3.1.4. Fuzzy based clustering to prune the size of the repository

The POF for most of the problems is very large and may even include an infinite number of individuals, and indeed the large number of Pareto-optimal solutions would mean a greater computational burden. Moreover, because there are always memory constraints, it is necessary to decrease the size of the repository without destroying the characteristics of the POF. In order to choose the best solutions amongst the non-dominated solutions (repository members + non-dominated solutions found in the current iteration), a fuzzy decision making policy is proposed to achieve a satisfactory plan [41]. Since the behavior of the objective functions conflicts, the Decision Maker (DM) can determine their preferences for corresponding fuzzy objectives in the fuzzy set to derive an efficient Pareto-optimal solution [41]. In this regard a linear membership function was calculated for each objective function:

$$\mu_{F_q}(X_i) = \begin{cases} 1, & F_q(X_i) \leqslant F_q^{min} \\ \frac{F_q^{max} - F_q(X_i)}{F_q^{max} - F_q^{min}}, & F_q^{min} \leqslant F_q(X_i) \leqslant F_q^{max} \\ 0, & F_q(X_i) \geqslant F_q^{max} \end{cases} \quad (8)$$

In order to generate a compromise value for each member of the repository, the DM is asked to specify the desired weight for each of objective function to represent the significance of each objective named as $w_{F_q}$, $q = 1, 2$. Then the following weighted and normalized membership approach can be used to extract better solutions in the premiere region with an iterative search which is expected to be the closest to the DM's requirements [40].

$$N_\mu(n_1) = \frac{\sum_{q=1}^{2} w_{F_q} \mu_{F_q}(n_1)}{\sum_{n_1=1}^{N_{rep}} \sum_{q=1}^{2} w_{F_q} \mu_{F_q}(n_1)} \quad (9)$$

Note that the value of $w_{F_q}$ was specified based on expert experience or the trial and error method.

In the multi-objective community detection problem, the POF was very large which means that each member of the repository is a cluster with a distinct radius, and adjacent clusters are joined together when the desired size repository meets. At the time, an individual with a higher weighted normalize membership from each cluster was selected to store in the repository.

#### 3.1.5. Smart population

The proposed algorithm is a non-linear, non-convex, non-smooth problem with a high dimension nature. Finding the optimal Pareto solutions with a uniform distribution for this problem is very difficult. Moreover, obtaining the best global solution in terms of a single objective problem is incentive enough to entice many researchers' to provide a technique powerful enough to solve these types of optimization problems. In this paper, to detect uniformly distributed POF and the extreme points, the EFA used a heuristic approach to select the population for the next iteration from solutions achieved through optimization. This approach comprised the two following phases:

*First phase*: It should be pointed out that the intention of the EFA was to find new non-dominated solutions when the repository was not filled. In this regard, non-dominated solutions are better for the next generation to encourage the fireflies to search for more Pareto-optimal solutions. Thus, the existing population was sorted based on the non-domination as follows: set $A_1$: solutions which are dominated one-time; set $A_2$: two-time dominated; set $A_3$: three-time dominated; etc., until all the populations are classified. Then the population of the next generation is determined with this priority list where the first non-dominated solutions are stored in the repository, followed by solutions for the set $A_1, A_2, A_3$, etc., until the number of individuals equals the size of the population.

*Second phase*: In this phase, the EFA tries harder than before to find near global solutions when the repository is filled. In this regard, $S$ percent of the repository are selected as the population and the remainder is randomly selected from the current population to be the population for the next iteration. This method optimizes the search procedure to find global and almost global solutions in a non-convex, non-smooth, non-linear, high dimension and high constraint problems. $S$ is defined as:

$$S = \left( \sin\left( \frac{\pi}{C} \times \frac{k_{\max} - k}{k_{\max} - k_{repo}} \right) \right) \times 100 \quad (10)$$

where $C$ is a constant value from $(0, 10]$. Variations in the range of $S$ are determined by $C$. The approximately linear characteristics to periodic one with several minimum and maximum for $S$ are illustrated in Fig. 1. Here, $C$ is 2 and $k_{repo}$ is 200. $k_{repo}$ is the latest iteration in which the repository has been filled.

### 3.2. Enhanced Firefly Algorithm (EFA)

#### 3.2.1. Firefly algorithm

FA is a an evolutionary algorithm based on population where searching for the optimum global solution is inspired by the behavior of social insects called fireflies, and the information exchanged is based on bio-luminescent communication [16]. FA is followed by three idealized rules to simplify its search process to achieve an optimal solution: (i) all the fireflies have the same sex and no mutation operation will be done to alter the attractiveness fireflies have for each other. However, in this paper, it is assumed the fireflies have a different sex where the mutation can be used between them, (ii) according to the other optimization algorithms, the

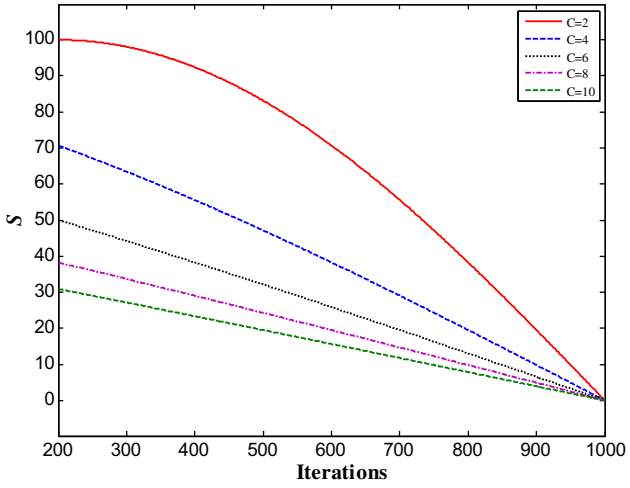


**Fig. 1.** Variations of $S$.

sharing of information or food between the fireflies is proportional to the attractiveness that increases with a decreasing Cartesian or Euclidean distance between them. If there is no brighter individual than a particular one in the problem search space, it will move randomly, and (iii) for maximization, the cost function is considered to be brightness of a firefly, but for minimization, the firefly with lowest fitness is the brightest of all the fireflies.

FA consists of a number of fireflies $X_m$, $m = 1,\ldots,N_{firefly}$) with different brightness $F(X_m)$. In order to take advantage of this technique, achieve a fitness function for each firefly and include the DM's preferences through the problem search spaces, the membership value of each firefly is determined according to (30). In addition, a reference membership value $\mu_{F_q}^{ref}$ is defined for each objective $q$. Reference membership values represent the importance of each objective and are specified based on expert judgment or trial and error.

The best solution, i.e.. the closest to the requirements of the DM, is the optimal solution to the following min–max problem [43]:

$$\min_{X_i\in\omega}\left\{\max_{q=1,2}\left[\mu_{F_q}^{ref}-\mu_{F_q}(X_i)\right]\right\} \tag{11}$$

where $\Omega$ is the set of solutions in the problem search space. In addition, each firefly $m$ at iteration $k$ is associated with a fitness value which is determined by the brightness $F(X_{i,m}^k)$ :

$$F\left(X_{i,m}^k\right)=\max_{q=1,2}\left\{\mu_{F_q}^{ref}-\mu_{F_q}\left(X_{i,m}^k\right)\right]\Big\} \tag{12}$$

In each iteration, and similar to the nature of the fireflies' republican nature, the firefly with the best fitness value is selected to participate in the next phase of optimization. The structure of each firefly in the group can be defined as follows:

$$F\left(X_{i,m}^k\right)=\left[F\left(X_{i,1}^k\right),\ldots,F\left(X_{i,NT}^k\right)\right],\quad i=1,\ldots,N_{firefly} \tag{13}$$

$NT$ is time horizon. The $m$th firefly is attracted to another brighter firefly $n$. This firefly modifies its position using the current position, the Cartesian or Euclidean distance from $F\left(X_{i,m}^k\right)$ to $F\left(X_{i,n}^k\right)$ which is brighter than it, and its random movement. The modified position is calculated according to the following equation [16]:

$$X_{i,m,firefly}^k=\begin{cases}X_{i,m,firefly}^k+\beta^k\left(X_{i,n,firefly}^k-X_{i,m,firefly}^k\right)+\alpha^k\left(rand_{1\times NT}(\cdot)-\frac{1}{2}\right),\\ \qquad\qquad \text{if } F\left(X_{i,n,firefly}^k\right)<F\left(X_{i,m,firefly}^k\right)\\ X_{i,n,firefly}^k,\quad \text{otherwise}\end{cases} \tag{14}$$

$$m=1,\ldots,N_{firefly};\quad n=1,\ldots,N_{firefly}$$

The attractiveness function $\beta^k$ can be expressed as follows [16]:

$$\beta^k=\beta_{\max}e^{-\gamma\left(r_{mn}^k\right)^2} \tag{15}$$

In this study, in order to improve $\beta^k$ such that a faster convergence of the algorithm and escape from local optima would be achieved, $\beta^k$ is rewritten as follows:

$$\beta^k=(\beta_{\max}-\beta_{\min})e^{-\gamma\left(r_{mn}^k\right)^2}+\beta_{\min} \tag{16}$$

where $r_{mn}^k$ is the Cartesian or Euclidean distance between $X_{i,n,firefly}^k$ and $X_{i,m,firefly}^k$. $\beta_{\max}$, $\gamma$ are commonly set to one [16]. $\beta_{\min}$ is a constant value which is fixed as 0.2.

In comparison with the other evolutionary algorithms, using FA has many major advantages in solving complex non-linear optimization problems. Some advantages are simple concepts, easy implementation, higher stability mechanism, and less execution. Despite having these features, it often experiences inappropriate convergence due to the local optima, lack of diversity of the fireflies, or a slow algorithm, all of which would be corrected as described next.

*3.2.2. Self-adaptive probabilistic mutation strategy*

In order to improve the performance of the original FA and escape from local optima, a new mutation strategy with two methods proposed in this study to improve the solutions, like the mutation in GA this mutation helps the firefly algorithm to escape from local optimal. All the fireflies in the population will have a chance to mutate, controlled by the probability of their methods of mutating. The incident of mutation is guided by the requirements of the search according to the following equation, which adapts itself, and makes the mutation strategy an extremely useful tool. Based on a probability model, each firefly selects one of these methods. This probability model is based on the brightness of each firefly to provide more optimal solutions. In the proposed EFA, the probability of both mutation methods is first assumed to be $Pmut_{method} = 0.5$, $method = 1,\ 2$ and a parameter is appropriated for each of these methods, named accumulator, as $a_{method} = 0$, $method = 1,\ 2$.

In each iteration the population is sorted according to the fitness function shown in (3). The brighter firefly is in the $m = 1$ place and the gloomy one is $m = N_{firefly}$. Then, the weight factor ($ww_m$) is labeled to each of them. The brighter firefly corresponds to the higher value of $ww_m$. In the proposed approach, $ww_m$ are calculated as follows:

$$ww_m=\frac{\log(N_{firefly}-m+1)}{\log(1)+\ldots+\log(N_{firefly})},\quad m=1,\ldots,N_{firefly} \tag{17}$$

The accumulator of each moving strategy is updated as:

$$a_{method}=a_{method}+\frac{ww_{mm}}{N_{method}},\quad mm=1,\ldots,N_{method} \tag{18}$$

where $N_{method}$ is the number of fireflies which chose the $method$th strategy for mutation and $ww_{mm}$, $mm = 1,\ldots,N_{method}$ are the weight factors corresponding to them. The excitation probability was formulated on the above discussion as:

$$Pmut_{method}=(1-\sigma)Pmut_{method}+\sigma\frac{a_{method}}{k_{\max}},\quad method=1,2 \tag{19}$$

where $\sigma$ is a rate to control the learning speed in the EFA algorithm, and $\sigma$ is set to 0.15 in this study. The values of normalized probability for each method of mutation are calculated as follows:

$$Pmut_{method}=\frac{Pmut_{method}}{Pmut_1+Pmut_2},\quad method=1,2 \tag{20}$$

Finally, the two proposed mutation strategies are described as follows:

*Mutation method 1*: This mutation technique is devised to implement the information from the best compromise solution found by the algorithm up to now as follows:

$$X_{i,mm,mut}^k=X_{i,mm,firefly}^k+rand_1(\cdot)(Best^k-M^k),\quad mm=1,\ldots N_1 \tag{21}$$

where $\mathbf{M}^k$ is the mean value of the population which can be calculated as:

$$\mathbf{M}^k=[\mathbf{me}_1^k,\ldots,\mathbf{me}_{NT}^k]$$
$$me_t^k=\frac{X_{1,t}^k+X_{2,t}^k+\cdots+X_{N_{firefly},t}^k}{N_{firefly}},\quad t=1,\ldots,NT \tag{22}$$

*Mutation method 2:* This method of mutation is proposed to improve the diversity of the solutions, alleviate stagnation and avoid being trapped in local optima. In each iteration and for each of the existing solutions, three vectors $r_1$, $r_2$, and $r_3$ were randomly

selected from the existing population to uniformly cover the algorithm searching domain. However, for the three vectors with limitation $r_1 \neq r_2 \neq r_3 \neq mm$, a mutant individual $\left(X^k_{i,mm,mut}\right)$ was generated as follows:

$$X^k_{i,mm,mut} = X^k_{i,r_1,firefly} + rand_1(\cdot)\left(X^k_{i,r_2,firefly} - X^k_{i,r_3,firefly}\right), \quad mm = 1,\ldots N_2 \tag{23}$$

After determining the mutant vector for all of the fireflies, this vector was mixed with $X^k_{i,m,firefly}$ which generated $X^k_{i,m,new}$ as:

$$X^k_{i,m,new} = \begin{cases} X^k_{i,m,mut} & \text{if } (rand_1 < rand_2) \\ X^k_{i,m,firefly} & \text{otherwise} \end{cases}$$

$i = 1,2,\ldots,N_{firefly}$ The new solutions can replace the original solution based on their fitness functions as follows:

$$X^{k+1}_{i,m,firefly} = \begin{cases} X^k_{i,m,new} & \text{if } F\left(X^k_{i,m,new} \leqslant X^k_{i,m,firefly}\right) \\ X^k_{i,m,firefly} & \text{otherwise} \end{cases} \tag{24}$$

After determining $X^{k+1}_{i,m,firefly}$, the fitness function $F\left(X^{k+1}_{i,m,firefly}\right)$ was compared to $F\left(X^k_{i,m,firefly}\right)$. If this solution is better than $X^{k+1}_{i,m,firefly}$ then it is replaced. This technique also enabled the EFA to provide better optimal solutions with respect to the DM's preferences, and significantly affected the convergence capability of the algorithm.

### 3.2.3. Chaotic formula for $\alpha$ in FA (CFA)

For a better performance of FA the parameters $\alpha$ and $\gamma$ should be carefully tuned. The random movement factor ($\alpha$) was selected in the range [0,1]. This coefficient affects the performance of FA. A large $\alpha$ makes the movement explore a large search area for the solution and a small $\alpha$ tends to facilitate a local search. The absorption coefficient ($\gamma$) controls the decrease in the intensity of the light and is commonly set to one, as suggested in [16]. Thus, these parameters affect the performance of the algorithm, so tuning them by a suitable method can enhance the ability of the FA. In this paper $\alpha$ was dynamically tuned at each iterations as:

$$c\alpha^k = \alpha^k D^k \tag{25}$$

where $c\alpha^k$ is the chaotic random movement factor at iteration $k$, $\alpha^k$ is the random movement factor which decreased linearly from 0.5 down to 0, and $D^k$ is the chaotic parameter at iteration $k$. The value of $D^k$ was determined by a iterator chaotic system, namely the logistic map, as follows [18]:

$$D^k = \mu \cdot D^{k-1}\left(1 - D^{k-1}\right) \tag{26}$$

where $\mu$ displays a control parameter which was selected between [0,4]. Variations of the $\mu$ were greatly influenced on the $D^k$ which was able to evaluate the $D^k$ at a constant size, and oscillated between a limited sequence of sizes, or oscillated randomly. Eq. (26) was without any stochastic pattern in the value of $\mu$ and indicated chaotic dynamic when $\mu = 4$ and $D^0 \notin \{0, 0.25, 0.50, 0.75, 1\}$. In this paper $D^0$ was equal to 0.54. The searching capability and efficiency of the proposed algorithm increased greatly with the chaotic $\alpha$ as illustrated in the numerical result. For instance, the performance of linearly decreased and chaotic-based $\alpha$ is illustrated in Fig. 2 for $k_{max} = 100$.

## 4. Application of EFA on the community detection problem

The process of the EFA can be summarized as follows and has been illustrated in Fig. 3:

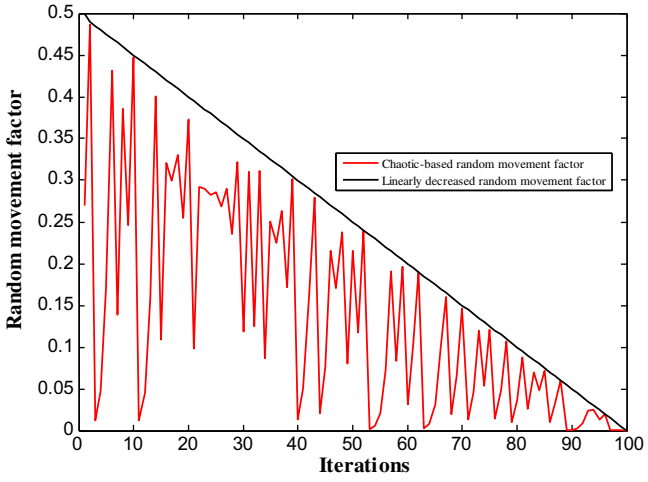


**Fig. 2.** Variations of $\alpha$.

*Step 1*: *Initializing the problem and algorithm parameters*: In this phase, as described above, we are interested in identifying a partitioning $\{C_1,\ldots,C_k\}$ that maximizes the number of connections inside each community and minimizes the number of links between the modules. The first objective was fulfilled by the community score. The first objective function is therefore

$$CS = \sum_{i=1}^{k} score(C_i) \tag{27}$$

The second objective is carried out by the community fitness by summing up the fitness of all the $C_i$ modules. The parameter $\alpha$, that tunes the size of the communities, has been set to 1 because in most cases the partitioning found for this value are relevant [31,44]. The second objective is therefore:

$$\sum_{i=1}^{k} P(C_i) \tag{28}$$

*Step 2*: *Representation*: Our partitioning algorithm uses the locus-based adjacency representation proposed in [45] and used by [46] for multi-objective clustering. In this graphic representation, an individual of the population consists of $N$ variable $x_1,\ldots,x_N$ and for each variable there is a set of possible range of values based on the adjacency matrix. For example, if node 1 has a connection with nodes 3, 5, and 6, the possible range of values for $x_1$ will be {3, 5, and 6}. For the isolated node k in the network, the possible range of values can be $\{1,2,\ldots,k-1, k+1,\ldots,N\}$.

Variables and values represent nodes of the graph $G = (V, E)$ modeling a network $N$, and a value $j$ assigned to the $i$th variable is interpreted as a link between the nodes $i$ and $j$ of $V$. This means that in the clustering solution found, $i$ and $j$ will be in the same cluster. However, a decoding step is needed to identify all the components of the corresponding graph. The nodes participating with the same component are assigned to one cluster. As observed in [46], the decoding step can be done in linear time. The main advantage of this representation is that the number of communities will be automatically determined by the number of components contained in an individual, and will be determined by the decoding step.

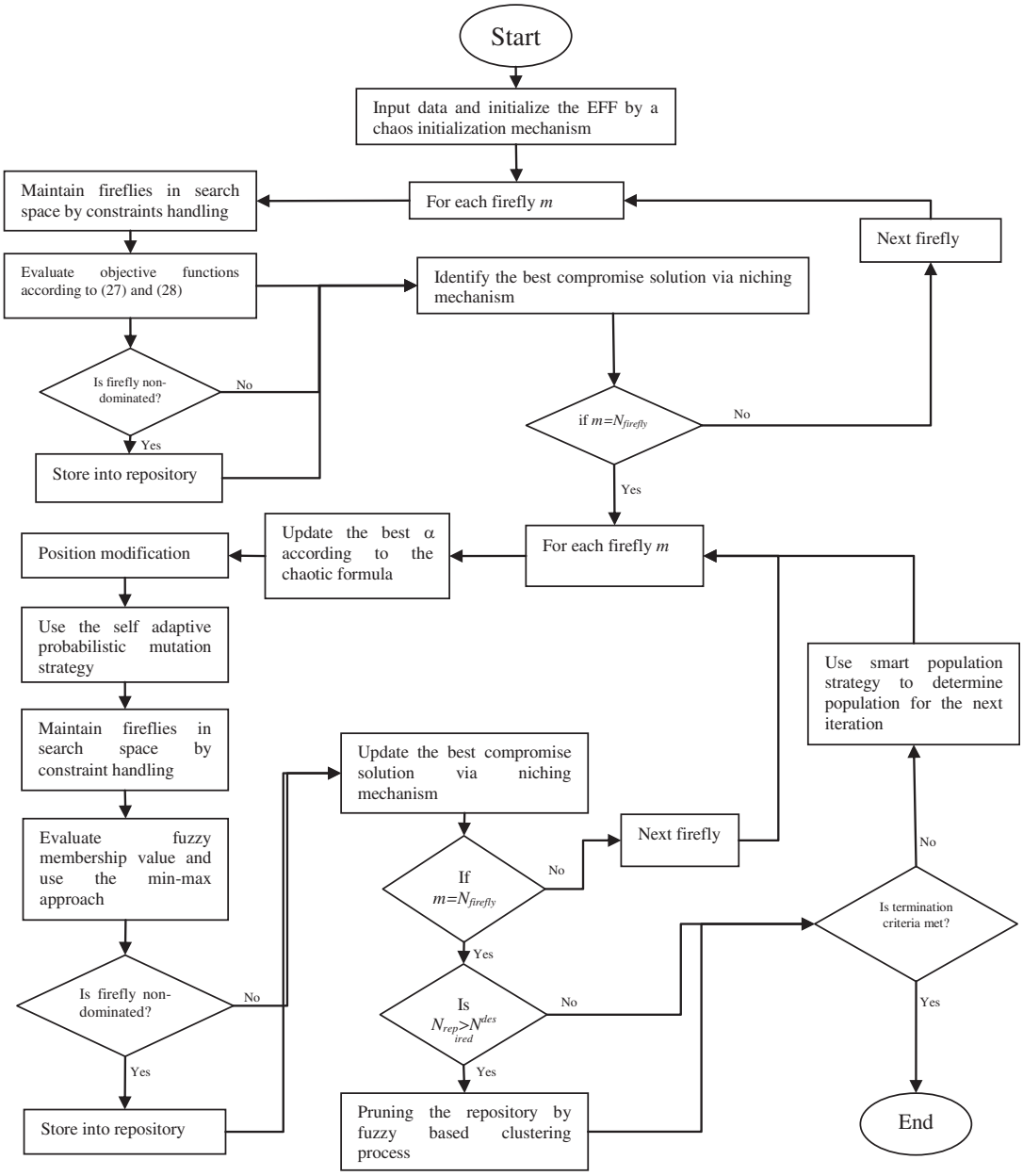


**Fig. 3.** Flowchart of the proposed EFA.

*Step 3*: *Generate an initial population by a chaos initialization*: An initial population based on the chaos initialization is used in this paper as follows:

$$Population = \begin{bmatrix} X_{i,1,firefly} \\ X_{i,2,firefly} \\ \vdots \\ X_{i,N_{firefly},firefly} \end{bmatrix} = \begin{bmatrix} X_{1,1,firefly} & \dots & X_{1,NT,firefly} \\ X_{2,1,firefly} & \dots & X_{2,NT,firefly} \\ \vdots & \ddots & \vdots \\ X_{N_{firefly},1,firefly} & \dots & X_{N_{firefly},NT,firefly} \end{bmatrix} \quad (29)$$

At the beginning the first firefly of the population was generated randomly from [0,1]. Then the following equation could be used for the rest of the population [18]:

$$X_{i,m,firefly} = \mu \times X_{i,m-1,firefly} \times (1 - X_{i,m-1,firefly}), \quad m = 2, \dots, N_{firefly} \quad (30)$$

*Step 4*: *Constitute the repository*: Store the non-dominated solutions and save them in the repository.

*Step 5*: *Reduce the repository size*: Use the fuzzy based clustering as described in Section 3.1.4 to achieve the desired size repository.

*Step 6*: *Update* $\mathbf{Best}^k$: In the current iteration ($k$), the niching mechanism was applied to the set of non-dominated solutions in the repository and the best compromise solution is selected as ($\mathbf{Best}^k$).

*Step 7*: *Update the EFA parameters*: The random movement factor $\alpha^k$ was updated by the chaotic sequence procedure described in Section 3.2.3.

*Step 8*: *Position modification*: To modify the position of each firefly, $\beta^k$ from (16) must be calculated and then each element in each individual moved toward the brighter one by (14).

*Step 9*: *Self-adaptive probabilistic mutation strategy*: A mutation strategy was performed on all the existing solutions based on Section 3.2.2.

*Step 10*: *Min_max approach*: Evaluate fuzzy membership values, compare with previous individuals and select the better individuals for being participated in the next step through (12).

*Step 11*: *Update the repository*: Check for non-domination. Update the repository.

*Step 12*: *Smart population*: Determine the population for the next iteration (Section 3.1.5).

*Step 13*: *Check convergence criteria*: Go to step 5 for the next iteration. This loop can be terminated after a predefined number of iterations and the best firefly with the best position is selected as the best compromise solution.

## 5. Experimental results

In this section we compare the effectiveness of the proposed multi-objective enhanced firefly with Clauset, Newman and Moore (CNM) [26], Rosvall and Bergstrom (RB) [47], Blondel et al. [27], Ronhovde and Nussinov (RN) [32], MOGA-Net [10] and the original Firefly Algorithm (FA) [17] using some real world datasets and synthetic benchmark datasets.

The effectiveness of stochastic algorithms is greatly dependent on the generation of initial solutions and therefore, for every dataset, algorithms have individually performed 100 times to test their own effectiveness, and each time with randomly generated initial solutions. Our algorithm was implemented into Matlab 7.1. All the experiments are conducted on a computer with Intel Core 2 Duo, 2.66 GHz, 4 GB RAM.

The settings of the proposed algorithm are as follows: Number of populations is set to 50 and 100. $k_{max}$ is 150 and 700. The attractiveness parameter $\beta^k$ is varied from $\beta_{min}$ to $\beta_{max}$. The absorption parameter $\Upsilon$ is set to one. The random movement factor $\alpha^k$ is dynamically tuned during the search process. In the case of EFA algorithm, based on the simulation results obtained for different values of $\beta_{min}$ and $\beta_{max}$, $\beta_{min} = 0.2$ and $\beta_{max} = 0.8$ gives optimal results. Also, after conducting a series of experiments, the value of $r_{share}$ is taken as 0.2. The $N_{rep}^{desired}$ is fixed to 100.

### 5.1. Evaluation criteria

Many methods have been developed for community detection till now, but it is still not clear which algorithms are reliable and shall be used in applications. The question of the reliability itself is tricky, as it requires shared definitions of community and partition which are, at present, still missing. This essentially means that, despite the huge literature on the topic, there is still no agreement among scholars on what a network with communities looks like.

We used Normalized Mutual Information (NMI) and Modularity (Q) to evaluate the quality of the proposed community detection method. The Normalized Mutual Information (NMI) is a similarity measure proven by Danon et al. [48] to be reliable. Given two partitions $A$ and $B$ of a network in communities, let $C$ be the confusion matrix whose element $C_{ij}$ is the number of nodes of community $i$ of the partition $A$ that are also in the community $j$ of the partition $B$. The normalized mutual information $I(A, B)$ is defined as:

$$I(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} C_{ij}\log\left(\frac{C_{ij}N}{C_{i.}C_{.j}}\right)}{\sum_{i=1}^{c_A} C_{i.}\log\left(\frac{C_{i.}}{N}\right) + \sum_{j=1}^{c_B} C_{.j}\log\left(\frac{C_{.j}}{N}\right)} \tag{31}$$

where $c_A(c_B)$ is the number of groups in the partition $A$ ($B$), $C_{i.}(C_{.j})$ is the sum of the elements of $C$ in row $i$ (column $j$), and $N$ is the number of nodes. If $A = B$, $I(A, B) = 1$. If $A$ and $B$ are completely different then $I(A, B) = 0$.

The modularity of Newman and Girvan [23] is a well known quality function used to evaluate the goodness of a partition. Let $k$ be the number of modules found inside a network, the modularity is defined as:

$$Q = \sum_{s=1}^{k}\left[\frac{l_s}{m} - \left(\frac{d_s}{2m}\right)^2\right] \tag{32}$$

where $l_s$ is the total number of edges joining vertices inside the module $s$, and $d_s$ is the sum of the degrees of the nodes of $s$. The first term of each summand of the modularity $Q$ is the fraction of edges inside a community and the second one is the expected value of the fraction of edges that would be in the network if they fell at random without regard to the community structure. Values approaching *1* indicate a strong community structure.

### 5.2. Real world networks

The Zackary's Karate Club network was generated by Zachary, who studied the friendship of 34 members of a karate club over a period of 2 years [49]. During this period, because of disagreements, the club divided in two groups almost of the same size.

*The Bottlenose Dolphins network*: A network of 62 bottlenose dolphins living in Doubtful Sound, New Zealand, was compiled by Lusseau after studying their behavior for 7 years. A tie between two dolphins was established by their statistically significant frequent association. The network split naturally into two large groups where the number of ties was 159 [50].

*The American College Football network*: Comes from the United States college football. The network represents the schedule of Division I games during the 2000 season. Nodes in the graph represent teams and edges represent the regular season games between the two teams they connect. The teams are divided into conferences. On average the teams played four inter-conference matches and seven intra-conference matches, thus they tend to play between members of the same conference. The network consists of 115 nodes and 616 edges grouped in 12 teams [20].

*The political books compiled by V. Krebs*: The nodes represent 105 books on American politics brought from Amazon.com, and the edges join pairs of books frequently purchased by the same buyer. Books were divided by Newman [51] according to their political alignment (conservative or liberal), except for a small number (13) having no clear affiliation.

### 5.3. Synthetic networks

We also used the Lancichinetti–Fortunato–Radicchi (LFR) benchmark graphs [44,52] to evaluate the performance of our algorithm. By varying the parameters of the networks we could analyze the behavior of the algorithms in detail. Some important parameters of the benchmark networks are given in Table 9. We generated several un-weighted undirected benchmark networks where the number of nodes $n = 5000$ and 50,000. For each n, two individual networks were generated with different ranges of community sizes, where *S* means that the sizes of the communities in the dataset were relatively small and *B* means that the sizes of the communities were relatively big. For each type of dataset, we ranged the mixing parameter mu from 0.1 to 0.8 with a span of 0.05 and got fifteen networks. Generally, the higher the mixture parameter of a network, the more difficult it is to reveal the structure of the community. Some important parameters of the benchmark networks are:

**Table 1**
NMI result obtained by the three algorithms on Zackary's Karate Club data.

| Method | NMI | | | Standard deviation |
|---|---|---|---|---|
| | Best | Average | Worst | |
| EFF | 0.9983 | 0.9983 | 0.9983 | 0.0000 |
| FA | 09803 | 0.9785 | 0.9772 | 0.0012 |
| RB | 0.9801 | 0.9780 | 0.9759 | 0.0021 |
| Blondel | 0.9901 | 0.9884 | 0.9867 | 0.0017 |
| RN | 0.9877 | 0.9875 | 0.9873 | 0.0002 |
| CNM | 0.6921 | 0.6675 | 0.6364 | 0.0270 |
| MOGA-Net | 0.9986 | 0.9986 | 0.9986 | 0.0000 |

**Table 2**
Modularity result obtained by the three algorithms on Zackary's Karate Club data.

| Method | Modularity | | | Standard deviation |
|---|---|---|---|---|
| | Best | Average | Worst | |
| EFF | 0.4201 | 0.4201 | 0.4201 | 0.0000 |
| FA | 0.4185 | 0.4174 | 0.4163 | 0.0011 |
| RB | 0.4187 | 0.4187 | 0.4187 | 0.0000 |
| Blondel | 0.4186 | 0.4175 | 0.4164 | 0.0011 |
| RN | 0.4177 | 0.4157 | 0.4137 | 0.0020 |
| CNM | 0.4176 | 0.4067 | 0.3948 | 0.0110 |
| MOGA-Net | 0.4151 | 0.4149 | 0.4148 | 0.0010 |

**Table 3**
NMI result obtained by the three algorithms on Bottlenose Dolphins data.

| Method | NMI | | | Standard deviation |
|---|---|---|---|---|
| | Best | Average | Worst | |
| EFF | 0.9888 | 0.9885 | 0.9884 | 0.0001 |
| FA | 0.9855 | 0.9852 | 0.9851 | 0.0002 |
| RB | 0.9857 | 0.9855 | 0.9853 | 0.0002 |
| Blondel | 09844 | 0.9843 | 0.9842 | 0.0001 |
| RN | .9879 | 0.9879 | 0.9879 | 0.0000 |
| CNM | 0.5744 | 0.5738 | 0.5732 | 0.0004 |
| MOGA-Net | 0.9996 | 0.9996 | 0.9996 | 0.0000 |

**Table 4**
Modularity result obtained by the three algorithms on Bottlenose Dolphins data.

| Method | Modularity | | | Standard deviation |
|---|---|---|---|---|
| | Best | Average | Worst | |
| EFF | 0.5242 | 0.5233 | 0.5231 | 0.0001 |
| FA | 0.5151 | 0.5149 | 0.5137 | 0.0002 |
| RB | 0.5149 | 0.5146 | 0.5143 | 0.0003 |
| Blondel | 0.5162 | 0.5152 | 0.5142 | 0.0010 |
| RN | 0.5186 | 0.5166 | 0.5146 | 0.0020 |
| CNM | 0.4961 | 0.4853 | 0.4774 | 0.0120 |
| MOGA-Net | 0.5048 | 0.5038 | 0.5029 | 0.0090 |

- $n$: number of nodes,
- $m$: average number of edges,
- $k$: average degree of the nodes,
- *maxk:* maximum degree,
- *mu:* mixing parameter, each node shares a fraction mu of its edges with nodes in other communities,
- *minc:* minimum for the community sizes,
- *maxc:* maximum for the community sizes.

### 5.4. Comparison of results

A comparison of results for each real world dataset is illustrated in Tables 1–8. And the evaluation results for synthetic datasets have been illustrated in Fig. 2.

**Table 5**
NMI result obtained by the three algorithms on American College Football data.

| Method | NMI | | | Standard deviation |
|---|---|---|---|---|
| | Best | Average | Worst | |
| EFF | 0.7988 | 0.7973 | 0.7959 | 0.0010 |
| FA | 0.7971 | 0.7938 | 0.7906 | 0.0033 |
| RB | 0.7964 | 0.7922 | 0.7880 | 0.0042 |
| Blondel | 0.7984 | 0.7962 | 0.7940 | 0.0022 |
| RN | 0.7943 | 0.7912 | 0.7881 | 0.0031 |
| CNM | 0.7619 | 0.7340 | 0.7170 | 0.0280 |
| MOGA-Net | 0.7953 | 0.7780 | 0.7640 | 0.0161 |

**Table 6**
Modularity result obtained by the three algorithms on American College Football data.

| Method | Modularity | | | Standard deviation |
|---|---|---|---|---|
| | Best | Average | Worst | |
| EFF | 0.6045 | 0.6023 | 0.6011 | 0.0018 |
| FA | 0.6011 | 0.5974 | 0.5936 | 0.0037 |
| RB | 0.5999 | 0.5938 | 0.5877 | 0.0061 |
| Blondel | 0.6017 | 0.5991 | 0.5965 | 0.0026 |
| RN | 0.6028 | 0.5988 | 0.5948 | 0.0040 |
| CNM | 0.5766 | 0.5474 | 0.5283 | 0.0294 |
| MOGA-Net | 0.5148 | 0.4978 | 0.4784 | 0.0158 |

**Table 7**
NMI result obtained by the three algorithms on Krebs' books on American politics data.

| Method | NMI | | | Standard deviation |
|---|---|---|---|---|
| | Best | Average | Worst | |
| EFF | 0.5990 | 0.5958 | 0.5953 | 0.0005 |
| FA | 0.5885 | 0.5871 | 0.5857 | 0.0014 |
| RB | 0.5987 | 0.5938 | 0.5877 | 0.0002 |
| Blondel | 0.5946 | 0.5991 | 0.5965 | 0.0011 |
| RN | 0.5973 | 0.5988 | 0.5948 | 0.0030 |
| CNM | 0.5296 | 0.5044 | 0.4792 | 0.0243 |
| MOGA-Net | 0.5971 | 0.5826 | 0.5675 | 0.0142 |

**Table 8**
Modularity result obtained by the three algorithms on Krebs' books on American politics data.

| Method | Modularity | | | Standard deviation |
|---|---|---|---|---|
| | Best | Average | Worst | |
| EFF | 0.5285 | 0.5285 | 0.5285 | 0.0000 |
| FA | 0.5185 | 0.5180 | 0.5175 | 0.0005 |
| RB | 0.5199 | 0.5197 | 0.5195 | 0.0002 |
| Blondel | 0.5207 | 0.5206 | 0.5205 | 0.0001 |
| RN | 0.5221 | 0.5215 | 0.5209 | 0.0006 |
| CNM | 0.5024 | 0.4818 | 0.5011 | 0.0177 |
| MOGA-Net | 0.5176 | 0.5136 | 0.5075 | 0.0039 |

**Table 9**
The parameters of the computer generated dataset for performance evaluation.

| Dataset | $n$ | $m$ | $k$ | maxk | minc | maxc |
|---|---|---|---|---|---|---|
| 5000S | 5000 | 48,811 | 20 | 50 | 10 | 50 |
| 5000B | 5000 | 49,009 | 20 | 50 | 20 | 100 |
| 50000S | 50,000 | 989,737 | 40 | 100 | 50 | 100 |
| 50000B | 50,000 | 990,687 | 40 | 100 | 100 | 200 |

The simulation result given in Tables 1–8 show that EFF is very precise. In other word, it provides the optimum value and small standard deviation in compare to those of obtained by the other

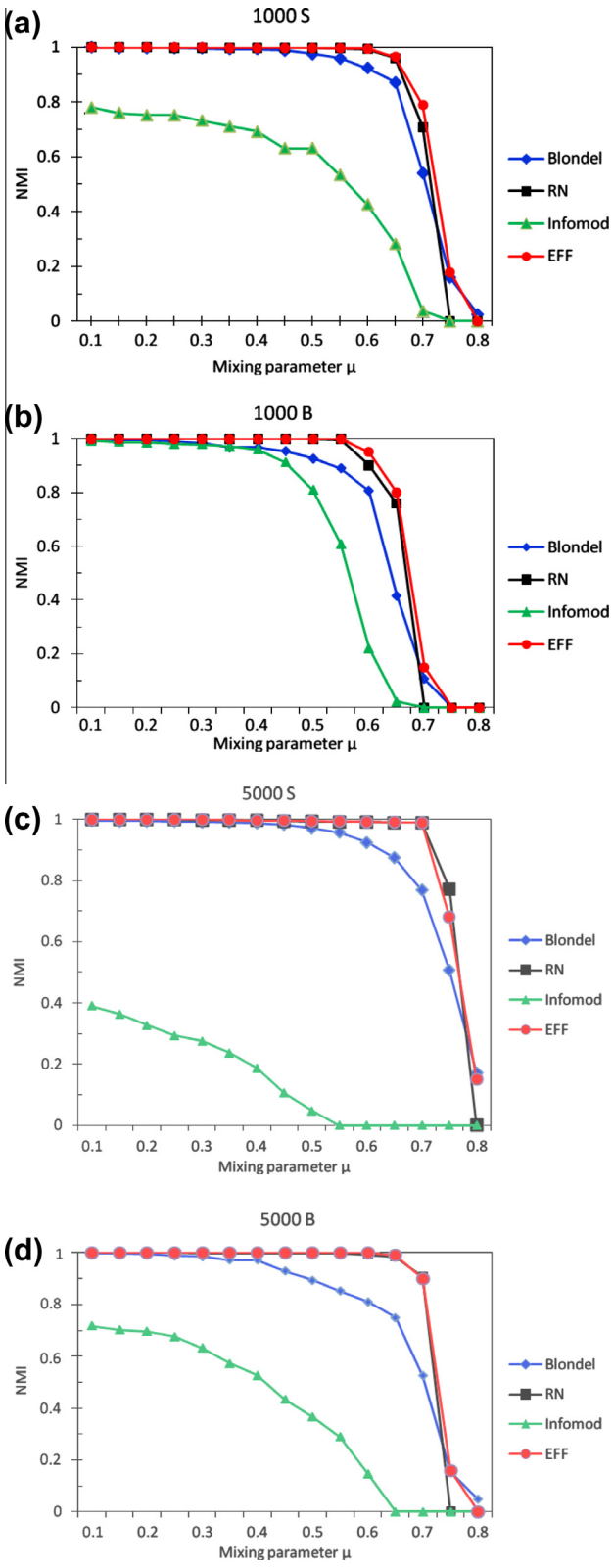


**Fig. 4.** Test of the accuracy of EFF, Bondel, Infomod and RN algorithms on the computer-generated benchmark networks.

methods. For instance, the results obtained on the Zachary's Karate Club dataset (Tables 1 and 2) show that the EFF converges to the global optimum of 0.9983 for *NMI* in all of runs and the Bondel reaches to 0.9901 at almost times while the best solution of FA, RB, RN, CNM and MOGA-Net are 0.9803, 0.9801, 0.9877, 0.6921 and 0.9986 respectively. The standard deviation of RN and MOGA-Net algorithms are 0.002 and 0.000, respectively, which they are smaller than other methods.

Table 2 shows the results of algorithms on Zachary's Karate Club dataset in terms of modularity, the optimum value is 0.4201, which is obtained in all the runs of the EFF algorithm. Noticeably other algorithms fail to attain this value even once within 100 runs. The FA, RB, Blondel, RN, CNM and MOGA-Net algorithms attained 0.4185, 0.4187, 4186, 0.4177, 0.4176 and 0.4151 respectively. The EFF algorithm was successful in finding four communities in all runs as well.

Tables 3 and 4 provide the result of algorithms on the Bottlenose Dolphins data. As seen from the results, the EFF algorithm the best values of 0.9888 and 0.5242 for *NMI* and modularity, respectively with standard deviation of 0.0001for both of them. In all the runs, four communities were detected by the EFF. The best *NMI* and modularity values provided by MOGA-Net were 0.9996 and 0.5048.

For the American College Football data (Tables 5 and 6) the best modularity, the worst modularity, the average modularity the EFF are 0.6045, 0.6023 and 0.6011 respectively with the standard deviation of 0.0018. Regarding the NMI the EFF algorithm found values of 0.7988, 0.7973 and 7959 for the best, worst and average global solutions with standard deviation of 0.0010. The EFF algorithm detected 11 communities for the American College Football data.

The FA, RB, Blondel, RN, CNM and MOGA-Net algorithms provided the best values of 0.7971, 0.7964, 0.7984, 0.7943, 0.7619 and 0.7953 for NMI and 0.6011, 0.5999, 0.6017, 0.6028, 0.5766 and 0.5148 in terms of modularity with standard deviation of 0.0033, 0.0042, 0.0022, 0.0031, 0.0280 and 0.0161.

Finally Tables 7 and 8 show the result of algorithms on the Krebs' books on the American Politics data that the EFF provided an optimum value of 0.0.5990 for *NMI* and standard deviation of 0.0005 and the modularity of 0.5285 in all runs. Four communities have been detected by the EFF algorithm in every solution.

The simulation results of the tables illustrate that the modularity and NMI the proposed EFF algorithm converge to the global optimum with a smaller standard deviation. The results illustrate that the proposed EFF community detection approach can be considered as a viable and an efficient heuristic to find optimal or near optimal solutions to the problem of community detection in networks.

We compare our algorithm with Bondel, Infomod and RN algorithms in term of NMI on synthetic datasets. The NMI scores of the four methods are plotted in Fig. 4. On most of the benchmark datasets, our algorithm gets NMI = 1 when $mu < 0.5$, which mean a perfect match with the original network structure.

As shown in Fig. 4 the EFF and RN algorithms get the value of 1 for NMI when $mu < 0.5$ that is much more better than the Blondel and Infomod algorithms and for $mu > 0.5$ the performance of the EFF and RN are almost the same or only slightly deferent in all generated networks.. In our experiments as illustrated in Fig. 4c and d the performance of our algorithm is decreased when $mu > 0.5$, especially in the small-scale network with big communities (e.g. 5000B).

We can see that the performance of EFF is better than that of Bondel, Infomod and RN algorithms on the generated networks in most cases.

## 6. Conclusion

We proposed a multi-objective method resolved a trade-off between the conflicting objectives of the community detection problem through the Pareto based approach which used the novel evolutionary algorithm named EFA. In order to strengthen the

proposed approach, various established heuristic techniques like FA, external repository, niching, fuzzy-based clustering, chaotic-based tuning of the algorithm parameter, self-adaptive probabilistic mutation strategy, and smart population approach were implemented with the proposed improvement. Also, the DM can update the solutions based on their preferences during simulation in two locations: i.e. changing the weighting factor and the reference membership functions of objectives. The proposed algorithm for community detection can be used when the number of clusters is unknown a priori. To evaluate the performance of the proposed algorithm, it was compared with the Bondel, Infomod, RN, MOGA-Net, CNM and original FF algorithms. The algorithm was implemented and tested on several real world and synthetic datasets, and showed that it was quite efficient at discovering the community structure of complex networks. Thus, this proposed algorithm can be considered as a viable and an efficient heuristic to find the optimal or near optimal solutions for community detection in social networks. The proposed EFF algorithm is not suitable for finding overlapping communities that could be extended in the future. Most of research in community detection is focused on static networks but almost all real networks are dynamic in nature. Detecting communities in dynamic networks is very challenging and the analysis of dynamic communities is still in its infancy and can be addressed by multiobjective evolutionary algorithm like the EFF algorithm in the future.

## References


[1] Z. Zhao, S. Feng, Q. Wang, J.Z. Huang, G.J. Williams, J. Fan, Topic oriented community detection through social objects and link analysis in social networks, Knowledge-Based Systems 26 (2012) 164–173.
[2] P. Liu, B. Raahemi, M. Benyoucef, Knowledge sharing in dynamic virtual enterprises: a socio-technological perspective, Knowledge-Based Systems 24 (2011) 427–443.
[3] W. Yuan, D. Guan, Y.K. Lee, S. Lee, S.J. Hur, Improved trust-aware recommender system using small-worldness of trust networks, Knowledge-Based Systems 23 (2010) 232–238.
[4] Z.Y. Xia, Z. Bu, Community detection based on a semantic network, Knowledge-Based Systems 26 (2012) 30–39.
[5] A. Ferligoj, V. Batagelj, Direct multicriteria clustering algorithms, Journal of Classification 9 (1992) 43–61.
[6] M. Tasgin, A. Herdagdelen, H. Bingol, Community Detection in Complex Networks using Genetic Algorithms, 2007. Arxiv preprint arXiv:0711.0491.
[7] X. Liu, D. Li, S. Wang, Z. Tao, Effective algorithm for detecting community structure in complex networks based on GA and clustering, Computational Science – ICCS 2007 (2007) 657–664.
[8] C. Pizzuti, Community detection in social networks with genetic algorithms, in: 10th annual conference on Genetic and evolutionary computation, 2008, pp. 1137–1138.
[9] C. Pizzuti, GA-Net: a genetic algorithm for community detection in social networks, Parallel Problem Solving from Nature – PPSN X (2008) 1081–1090.
[10] C. Pizzuti, A multi-objective genetic algorithm for community detection in networks, in: 21st International Conference on Tools with Artificial Intelligence, ICTAI '09, (2009) 379–386.
[11] D. He, Z. Wang, B. Yang, C. Zhou, Genetic algorithm with ensemble learning for detecting community structure in complex networks,in: Fourth International Conference on Computer Sciences and Convergence Information Technology, ICCIT '09, (2009) 702–707.
[12] J. Liu, T. Liu, Detecting community structure in complex networks using simulated annealing with *k*-means algorithms, Physica A: Statistical Mechanics and Its Applications 389 (2010) 2300–2309.
[13] A. Gog, D. Dumitrescu, B. Hirsbrunner, Community detection in complex networks using collaborative evolutionary algorithms, Advances in Artificial Life (2007) 886–894.
[14] J. Liu, W. Zhong, H.A. Abbass, D.G. Green, Separated, overlapping community detection in complex networks using multiobjective evolutionary algorithms, IEEE Congress on Evolutionary Computation (2010) 1–7.
[15] J. Kleinberg, An impossibility theorem for clustering, Advances in Neural Information Processing Systems (2003) 463–470.
[16] T. Apostolopoulos, A. Vlachos, Application of the firefly algorithm for solving the economic emissions load dispatch problem, International Journal of Combinatorics 2011 (2011) 1–23.
[17] X.S. Yang, Firefly algorithms for multimodal optimization, Stochastic Algorithms: Foundations and Applications (2009) 169–178.
[18] R. Caponetto, L. Fortuna, S. Fazzino, M.G. Xibilia, Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Transactions on Evolutionary Computation 7 (2003) 289–304.
[19] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell System Technical Journal 49 (1970) 291–307.
[20] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences 99 (2002) 7821.
[21] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 888–905.
[22] S. Smyth, A spectral clustering approach to finding communities in graphs, in: the Fifth SIAM International Conference on Data Mining, 2005, p. 27.
[23] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E 69 (2004) 026113.
[24] R. Guimera, L.A.N. Amaral, Functional cartography of complex metabolic networks, Nature 433 (2005) 895–900.
[25] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Physical Review E 69 (2004) 066133.
[26] A. Clauset, M.E.J. Newman, C. Moore, Finding community structure in very large networks, Physical Review E 70 (2004) 066111.
[27] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (2008) P10008.
[28] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proceedings of the National Academy of Sciences 104 (2007) 36.
[29] M. Sales-Pardo, R. Guimera, A.A. Moreira, L.A.N. Amaral, Extracting the hierarchical organization of complex systems, Proceedings of the National Academy of Sciences 104 (2007) 15224.
[30] A. Arenas, A. Fernandez, S. Gomez, Analysis of the structure of complex networks at different resolution levels, New Journal of Physics 10 (2008) 053039.
[31] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New Journal of Physics 11 (2009) 033015.
[32] P. Ronhovde, Z. Nussinov, Multiresolution community detection for megascale networks by information-based replica correlations, Physical Review E 80 (2009) 016109.
[33] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[34] T. Nepusz, A. Petróczi, L. Négyessy, F. Bazsó, Fuzzy communities and the concept of bridgeness in complex networks, Physical Review E 77 (2008) 016107.
[35] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, KDD 96, 1996, pp. 226–231.
[36] M. Ankerst, M.M. Breunig, H.P. Kriegel, J. Sander, OPTICS: ordering points to identify the clustering structure, in: ACM SIGMOD international conference on Management of data, 1999, pp. 49–60.
[37] X. Xu, N. Yuruk, Z. Feng, T.A.J. Schweiger, Scan: a structural clustering algorithm for networks, in: the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 824–833.
[38] D. Bortner, J. Han, Progressive clustering of networks using structure-connected order of traversal, 2010, pp. 653–656.
[39] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Proceedings of the National Academy of Sciences of the United States of America 101 (2004) 2658.
[40] M.A. Abido, Multiobjective evolutionary algorithms for electric power dispatch problem, Computational Intelligence (2009) 47–82.
[41] S. Agrawal, B. Panigrahi, M.K. Tiwari, Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch, IEEE Transactions on Evolutionary Computation 12 (2008) 529–541.
[42] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-wesley, 1989.
[43] M. Sakawa, H. Yano, An interactive fuzzy satisficing method for generalized multiobjective linear programming problems with fuzzy parameters, Fuzzy Sets and Systems 35 (1990) 125–142.
[44] A. Lancichinetti, S. Fortunato, Community detection algorithms: a comparative analysis, Physical Review E 80 (2009) 056117.
[45] Y.J. Park, M.S. Song, A genetic algorithm for clustering problems, in: 3rd Annual Conference on Genetic Algorithms, 1989, pp. 2–9.
[46] J. Handl, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE Transactions on Evolutionary Computation 11 (2007) 56–76.
[47] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proceedings of the National Academy of Sciences 105 (2008) 1118.
[48] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment 2005 (2005) P09008.
[49] W.W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research (1977) 452–473.
[50] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54 (2003) 396–405.
[51] V. Krebs. <http://www.orgnet.com/>.
[52] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Physical Review E 78 (2008) 046110.