

A discrete modified fireworks algorithm for community detection in complex networks

Mohamed Guendouz¹ · Abdelmalek Amine¹ · Reda Mohamed Hamou¹

Published online: 3 September 2016
© Springer Science+Business Media New York 2016

Abstract In the last decade, several algorithms have been proposed to solve the problem of community detection in complex networks. Many of them are based on swarm intelligence and evolutionary algorithms. Most of these algorithms use the modularity density as a fitness function to maximize. However, these algorithms attempt to find the best solution without taking into consideration the structure of the network. In this paper, a new discrete modified Fireworks Algorithm (FWA) has been developed to solve the problem of community detection. A new initialization strategy and new mutation strategies are proposed, based on the label propagation strategy to enhance the algorithm and to speed up its convergence. The proposed algorithm has been evaluated on real-world and synthetic networks. Experimental results compared with three other known algorithms show the effectiveness of using our proposed algorithm for solving the problem of detecting communities in complex networks.

Keywords Community detection · Evolutionary algorithms · Swarm intelligence · Fireworks algorithm

1 Introduction

Complex networks have known an enormous interest by researchers from different fields in recent years, since many real-world complex systems take the form of complex networks including: Internet, Biology, Physics, Neural Networks and Communications. Networks are presented as graphs where vertices represent individual objects and edges represent the interactions between them. Individual objects could be users of a social networking website, website pages or sensors, whereas interactions could be friendship relations, URL links or data communication respectively. Networks are usually found to group into strongly connected groups, with a high density of within-group edges and a lower density of between-group edges. This important property of networks is called community structure. Detecting these groups, also known as clusters, partitions or communities, is called community detection, which is one of the primary topics in the field of networks analysis. It helps to get a clear global view of individual objects interactions in the network.

Community detection in networks can be a difficult computational task; the communities have different sizes, and their number is usually unknown. Even with these difficulties, several methods and algorithms for community detection have been proposed and applied to many real world problems. Traditional methods can be classified into two types: Graph partitioning and Hierarchical clustering. The graph partitioning methods usually cluster a network into a predetermined number of communities with equal size. These methods require the number and the size of communities before partitioning. The Kernighan-Lin algorithm [33] is one of the earliest methods proposed for graph

✉ Mohamed Guendouz
med.phd.guendouz@gmail.com

Abdelmalek Amine
amine_abd1@yahoo.fr

Reda Mohamed Hamou
hamoureda@yahoo.fr

¹ GeCoDe Laboratory, Tahar Moulay University of Saida, Saida, Algeria

partitioning. The hierarchical clustering methods cluster a network into groups of nodes based on their similarity, which means similar nodes are grouped into communities according to a similarity measure. Cosine similarity, Hamming distance, and Jaccard index [26, 32] are frequently used measures, and these methods can be classified into two categories:

1. Agglomerative algorithms: Initially, each node represents a partition of its own, then partitions are successively merged until the desired network partition structure is obtained.
2. Divisive algorithms: All nodes initially belong to one partition, then the partition is divided into sub-partitions, which are successively divided into their sub-partitions. This process continues until the desired network partition structure is obtained.

Hierarchical methods have the advantage of not requiring a predefined number and size of partitions. However, they do not provide a way to choose the better partition that represents the community structure of the network from those obtained by the procedure. A detailed classification of community detection algorithms can be found in [7]

In the last decade, the problem of community detection has been extensively studied by many researchers from different domains, and several approaches and algorithms have been proposed to detect communities in complex networks. The Girvan-Newman algorithm is one of the most famous ones; it was proposed by Girvan and Newman in [10, 25]. This algorithm continuously removes edges from the original network, until no connected edges are found. The algorithm chooses edges to remove by using the betweenness measure. This measure counts the number of shortest paths from all nodes to all others that pass through that edge. At each iteration, the edge with the highest betweenness value is removed. This process is repeated until reaching a stop criterion. Newman presents the modularity as a stop criterion, which has been widely used later in many other methods.

The great success of different evolutionary and swarm intelligence optimization algorithms on solving various real world problems has influenced many researchers to apply them to community detection problems. The GA-Net is one of the first approaches which has been proposed by Pizzuti in [27]. This method uses a genetic algorithm to detect communities in complex networks by maximizing the community score as a single-objective optimization problem. Pizzuti also proposed a multi-objective optimization method to solve this problem in [28]. In recent years, many researchers have proposed several methods for detecting communities by maximizing the modularity value using different evolutionary algorithms, including [4, 11, 20, 21,

31, 36]. However, the modularity has been found to have many disadvantages. First, Brandes et al. [1] proves that the maximization of the modularity value is an NP-hard problem. Second, authors in [15] found that large modularity value does not mean necessarily better community structure. Third, Fortunato and Barthelemy [8] show the big disadvantage that modularity has a resolution limitation.

To overcome this limitation, the modularity density (denoted as D) has been proposed by Li [19]. The modularity density is a quantitative measure that uses the average modularity degree to evaluate a community structure; the larger the value of D , the more accurate a partition is. Authors also proved that maximizing modularity density could overcome the resolution limit of modularity. As a result, many algorithms have been proposed to find partitions in networks by maximizing the modularity density value including [3, 12, 13, 16]. A detailed survey on community detection using evolutionary algorithms can be found in [2].

The Fireworks Algorithm (FWA) is one of the latest swarm intelligence optimization algorithms, which simulates the explosion of fireworks and the generation of sparks in the sky. It was proposed originally by Tan [35] as an evolutionary algorithm for continuous optimization problems. In this paper, a novel discrete fireworks algorithm variant is proposed to solve community detection problems by maximizing the modularity density function. We propose a new initialization and mutation strategy based on the influence of node neighborhood.

The rest of this paper is organized as follows: Section 2 introduces the community detection problem and gives some basic notations used in this paper. Section 3 presents the fireworks algorithm in detail. Section 4 gives a detailed description of our proposed algorithm. Section 5 shows and discusses the obtained results from evaluating our proposed algorithm on synthetic and real-world networks, it also shows the comparison of the proposed algorithm with others. Finally, the conclusion is summarized in Section 6.

2 Community detection problem

This section presents the problem of community detection and gives some definitions.

A network can be presented by a graph $G(V, E)$, where $V = \{v_1, v_2 \dots v_n\}$ represents vertices or nodes and $E \subseteq V \times V$ represents edges. The network structure can be presented by a $n \times n$ adjacency matrix A . Each element A_{ij} can be either 1 or 0, $A_{ij} = 1$ if there is an edge e_{ij} between vertices v_i and v_j otherwise $A_{ij} = 0$. Assume that $S \subset G$ is a subgraph where node i belongs to it, $k_i^{in} = \sum_{i,j \in S} A_{ij}$ and $k_i^{out} = \sum_{i \in S, j \notin S} A_{ij}$ are respectively the internal and the

external degree of node i , then the community represented by the subgraph S has usually the following property:

$$\sum_{i \in S} k_i^{in} > \sum_{i \in S} k_i^{out} \tag{1}$$

This property means that the sum of all degrees within the community is larger than the sum of all degrees toward the rest of the network.

The community detection problem can be formulated as a modularity (Q) maximization problem. The modularity (Q) was proposed by Newman in [24] to evaluate a community structure of a network. Q is mathematically described by the following equation:

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(i, j) \tag{2}$$

Where k_i and k_j are respectively degrees of nodes i and j , m is the total number of nodes in the network. $\delta(i, j) = 1$ if nodes i and j are in the same community, otherwise 0. Large Q value means better community structure. Otherwise, the structure is more ambiguous. Li [19] proposed another mathematical form for writing the modularity, described as follows:

$$Q = \sum_{i=1}^N \left[\frac{L(V_i, V_i)}{L(V, V)} - \left(\frac{L(V_i, V)}{L(V, V)} \right)^2 \right] \tag{3}$$

Where N is the number of communities, $L(V_n, V_m) = \sum_{i \in V_n, j \in V_m} A_{ij}$, $L(V, V) = 2m$ and $L(V_i, V) = \sum_{j \in V, k \in V} A_{jk}$

The modularity has been used in many community detection methods as an objective function to maximize or as an evaluation criterion. Until, when Fortunato and Barthelemy [8] prove that the modularity has a resolution limitation. To overcome this limitation Li et al. [19] proposed the modularity density (D) which is based on the average modularity degree, and it is defined as follows:

$$D = \sum_{i=1}^N \frac{L(V_i, V_i) - L(V_i, \bar{V})}{|V_i|} \tag{4}$$

Where $\frac{L(V_i, V_i)}{|V_i|}$ and $\frac{L(V_i, \bar{V})}{|V_i|}$ represent the average internal and external degrees of the i th community, D aims to maximize the internal degree and minimize the external degree of the community structure.

In this paper, we propose a new method for detecting communities in complex networks based on the fireworks algorithm by maximizing the value of modularity density (D).

3 Fireworks algorithm (FWA)

Fireworks Algorithm (FWA) [35] is a metaheuristic swarm intelligence optimization algorithm proposed by Tan in 2010, initially inspired by the explosion of the fireworks and firecrackers in the sky, especially in Chinese spring festivals where many fireworks are triggered to generate sparks. The basic idea behind the fireworks algorithm is that the way fireworks explode in sky is similar to the way of searching for the optimal solution in swarm intelligence algorithms. When a firework explodes, a number of sparks are generated around it. The number and the amplitude of generated sparks change from one firework to another, according to their quality. Fireworks with good quality generate more sparks within a smaller amplitude, and fireworks with bad quality generate fewer sparks within a larger amplitude. Fig. 1 shows two examples of firework’s explosion, bad and good explosion.

Similar to other swarm intelligence algorithms, the fireworks algorithm comprises four parts: the explosion operator, the mapping rule, the mutation operator, and the selection strategy. The algorithm generates initially a population of N random fireworks, also called a population of individuals. Then, every firework in the population explodes and generates sparks around it. The number and the amplitude of these generated sparks are determined by the explosion operator. After that, a Gaussian mutation operator will be applied to every spark to keep the diversity of the population. Finally, the algorithm keeps the best individual in the population and selects the rest ($N - 1$) individuals for the next generation based on their distances. These parts and strategies are described in detail as follows:

3.1 Explosion operator

At the first iteration, the FWA generates N random fireworks, then, each firework generates sparks using explosion operations. The explosion operator plays a key role in the FWA algorithm. It is responsible for calculating the number

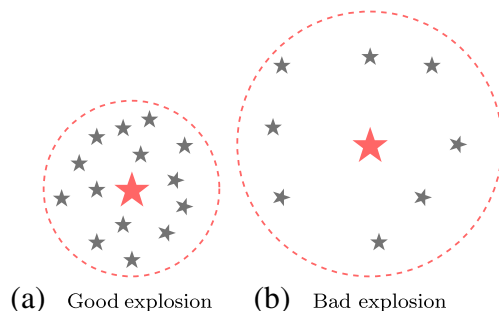


Fig. 1 Example showing two firework’s explosion

and amplitude of generated sparks. The explosion operator consists of two parameters; the first one is the number of sparks S_i and it is calculated using the following equation:

$$S_i = m \times \frac{Y_{max} - f(x_i) + \varepsilon}{\sum_{i=1}^N (Y_{max} - f(x_i)) + \varepsilon} \quad (5)$$

Where S_i is the number of sparks for each individual or firework, m is a constant represents the total number of sparks and Y_{max} is the fitness value of the worst individual among the N individuals in the population. Function $f(x_i)$ represents the fitness of individual x_i , while the last parameter ε is used to prevent the denominator from becoming zero.

The second parameter is the amplitude of sparks A_i ; it is calculated as follows:

$$A_i = \hat{A} \times \frac{f(x_i) - Y_{min} + \varepsilon}{\sum_{i=1}^N (f(x_i) - Y_{min}) + \varepsilon} \quad (6)$$

Where A_i denotes the amplitude of each individual, \hat{A} is a constant number that represents the sum of all amplitudes, while Y_{min} is the fitness value of the best individual among the N individuals. The meaning of function $f(x_i)$ and parameter ε are the same as mentioned in the previous definition.

The explosion amplitude will be used to determine the displacement of each generated spark around the exploded firework. FWA generates different random movements to ensure the diversity of the population based on the number of sparks and explosion amplitude of each firework. The displacement operation is defined as:

$$\Delta x_i^k = x_i^k + U(-A_i, A_i) \quad (7)$$

Where $U(-A_i, A_i)$ denotes the uniform random number within the intervals of the amplitude A_i , and x_i^k represents the k^{th} position of the i^{th} firework.

3.2 Mapping rule

The mapping rule is used to keep sparks inside the feasible search space, if a spark lies out the boundary, then its position will be mapped to a new feasible one inside the border. The mapping rule is defined as follows:

$$x_i = x_{min} + |x_i| \bmod (x_{max} - x_{min}) \quad (8)$$

Where x_i represents the i^{th} position of the spark, while x_{max} and x_{min} stand for the maximum and minimum boundary of position i .

3.3 Gaussian mutation operator

The Gaussian mutation operator is used to keep the diversity of the population and to make the algorithm search in

all the global search space, given a position of an individual denoted by x_i^k , the Gaussian operator is calculated as follows:

$$x_i^k = x_i^k \times g \quad (9)$$

Where g is a random number in Gaussian distribution.

$$g = \text{Gaussian}(1, 1) \quad (10)$$

3.4 Selection strategy

After applying the explosion operator, the mutation operator, and the mapping rule, the FWA selects the best generated sparks to be passed to the next generation. In FWA, the best spark is always kept and the rest $N - 1$ sparks are selected using a distance-based strategy. Usually, the Euclidean method is used to calculate distance between two sparks; it is denoted by d and defined as follows:

$$d(x_i, x_j) = |f(x_i) - f(x_j)| \quad (11)$$

Where $f(x_i)$ and $f(x_j)$ are fitnesses of individuals x_i and x_j respectively.

Finally, a roulette wheel method is used to calculate the possibility of selecting an individual.

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_j)} \quad (12)$$

Where $R(x_i) = \sum_{j=1}^K d(x_i, x_j)$ represents the sum of distances between individual x_i and all the other individuals, K is a set, contains all generated sparks as well as the N fireworks. Fig. 2 shows the flowchart of the fireworks algorithm.

Since its development in 2010, the fireworks algorithm has attracted the attention of many researchers from different fields, and it was successfully applied to solve many real-world problems such as TSP problem [34], data perturbation [29] and Oil crop production [42]. Many modifications and enhancements have been proposed to it including [18, 39–41, 43] and a GPU-based parallel implementation [6].

4 The proposed discrete fireworks algorithm

Since the fireworks algorithm is designed to deal only with real-valued optimization problems, it cannot be applied directly to solve the community detection problem, which is a discrete optimization problem. Therefore, we propose in this paper a novel discrete modified fireworks algorithm, named DMFWA, to solve this problem.

This section is organized as follows: First, the solution encoding format is given. Next, the objective function of the

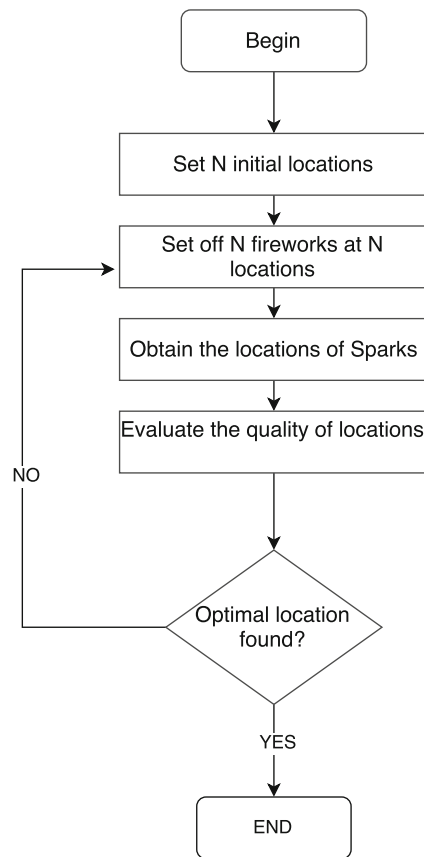


Fig. 2 Flowchart of the Fireworks Algorithm (FWA)

algorithm is presented. After that, our proposed initialization strategy and mutation operator are described. Finally, the framework of the proposed algorithm is elaborated.

4.1 Individual representation

Generally, in swarm intelligence algorithms, the representation of individuals, also called, the encoded form or solution encoding, has a significant impact on solving the problem itself. In FWA, a good representation of fireworks can reduce the search space and speed up the convergence of the algorithm. In this paper, we adopt the string-based representation. In this representation, the number of positions is equal to the number of nodes in the network. An arbitrary individual X_i represents the community structure of the network where each position X_i^k represents the label $L(k)$ of node k . For instance, if $X_i^1 = X_i^3 = 1$, then nodes 1 and 3 are in the same community labeled by 1, we can also write, $L(1) = L(3) = 1$. This representation schema is easier to code and decode. Also, the number of communities will be automatically determined, and will be equal to the number of unique labels.

An example of the solution encoding and its corresponding network is shown in Fig. 3. As indicated by Fig. 3a, the

network consists of six nodes numbered from 1 to 6. A possible optimal solution is given in Fig. 3b. Nodes 1 to 4 are clustered into the first community labeled by 1, while nodes 5 and 6 are in the second community labeled by 2. This solution is translated in the community structure shown in Fig. 3c.

4.2 Objective function

Several objective functions have been proposed to solve community detection problems. In this paper, the modularity density (D) is used as an objective function for our proposed algorithm. The modularity density has been explained in detail in Section 2.

4.3 Fireworks initialization

The initialization of the first population plays an important role; better initialization can generate good solutions and reduce the searching time. The FWA uses a random strategy to initialize the first population of fireworks; it does not take any prior knowledge about the specified problem into consideration. In this paper, we introduce a new fireworks initialization strategy that takes the original network structure into consideration. This approach is based on the influence of node's neighbors. The strategy is as follows:

1. Initialize a population of N fireworks (N is equal to 50 fireworks).
2. For each firework X_i , $X_i^k = k$; each node will be in its community.
3. For each firework, select a random node j , get node k from neighbors of j which has the highest degree. Assign all neighbors of k to the same community.

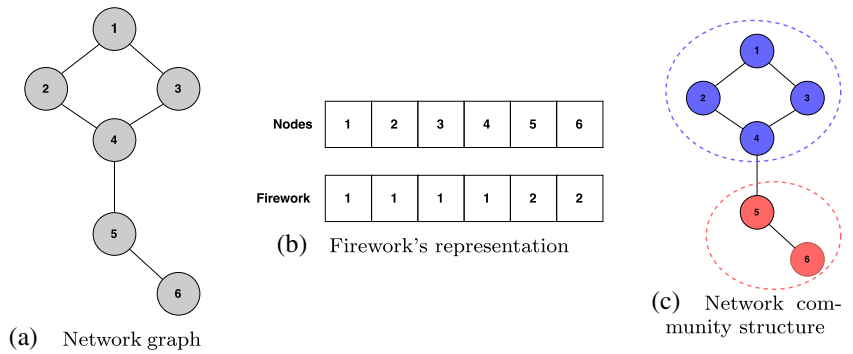
If the randomly chosen node j has no neighbors, the algorithm selects another one.

4.4 The mutation operator

The mutation is an important operation in the fireworks algorithm. It is responsible for updating individual's positions after each iteration. The displacement operator and the Gaussian mutation strategy are usually used to update individual's position in the original fireworks algorithm. However, these two operators have been designed initially to deal with continuous values, and they can not be used in our proposed algorithm where values are in a discrete search space.

In this paper, we propose a new mutation operator based on the label propagation strategy introduced in [14]. This strategy uses labels to assign each node to a community, and the nodes with the same label are considered to be in the same community. The label propagation strategy update

Fig. 3 Solution encoding



node's label, based on its neighborhood labels. The label which frequently appears in its neighborhood, is used as its new label.

The newly defined mutation operator is defined as follows:

$$x_i^k = \begin{cases} x_i^k & \text{if } \text{random}(0, 1) \geq \text{sigmoid}(A_i) \\ Nbest_j & \text{if } \text{random}(0, 1) < \text{sigmoid}(A_i) \end{cases} \quad (13)$$

Where x_i^k is the current k^{th} position of firework i , which is representing the current label of node k . $\text{random}(0, 1)$ is a randomly generated number between 0 and 1. $Nbest_k$ is the most common used label by neighbors of node k . $\text{sigmoid}(A_i)$ is the sigmoid function of the amplitude A_i of firework i . The amplitude A_i is calculated as in the original fireworks algorithm by using (6). The sigmoid function is defined as:

$$\text{sigmoid}(x) = \frac{1}{1 + \exp^{-x}} \quad (14)$$

An illustrative example of this newly defined mutation operator for discrete fireworks algorithm can be found in Fig. 4. In this figure, $X_i(t)$ represents the current location of firework X_i . The label of node 3 is 2, which means that it is in the community labeled by 2 with node 6. When updating the current firework location to the new location $X_i(t + 1)$, the label of node 3 is changed to 1 because the label 1 is the most frequently used one in its neighborhood nodes.

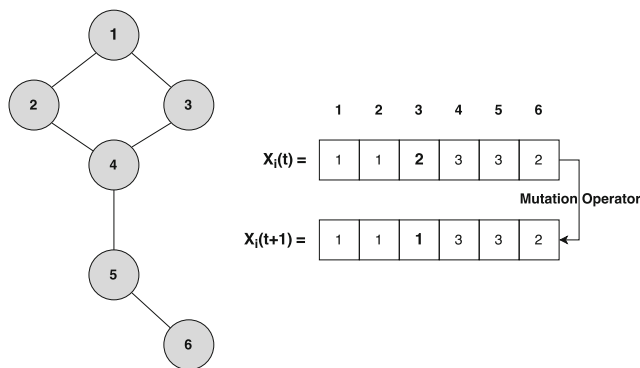


Fig. 4 Description of the proposed mutation strategy

4.5 Framework of the proposed algorithm

After describing all the strategies in detail, the complete framework of the proposed DMFWA for community detection in complex networks is given in Algorithm 1

Algorithm 1 Framework of the proposed algorithm DMFWA

```

Input:
Aij : The adjacency matrix of the network
N: Population size
Iter: Number of iteration
Output:
XBest: The firework with the best fitness value, which corresponds to the optimal network partition
Initialization
while CurrentIter < Iter do
  foreach Xi ∈ P do
    Explode firework Xi;
    Calculate number of sparks Si using 5;
    Calculate explosion amplitude Ai using 6;
    Generate sparks;
    foreach generated spark Xj do
      if rand(0, 1) < sigmoid(Ai) then
        Update the kth position of the spark using the proposed mutation operator 13;
        Xjk = Nbestk
      else
        Keep the kth position of the spark;
        Xjk = Xik
    Add spark to S. This set contains all generated sparks.
  Calculate the fitness value of each spark in set S and each firework of the population P using the modularity density D;
  Select the Best spark/firework that has the best fitness value;
  Select the rest N - 1 using the selection strategy;
  if Iterations done then
    return Best
    
```

5 Experimental results and discussion

This section presents and discusses the obtained results from evaluating our proposed method on synthetic and real-world networks. It also shows the results from comparing our method with the other three methods: CNM, Infomap, and GA-Net.

The CNM algorithm is a fast greedy modularity optimization method, which was proposed by Clauset et al. in [5]. This method is in fact a fast implementation of the original Girvan-Newman algorithm. This algorithm is abbreviated by CNM acronym which represents the first characters of author's names. The second algorithm is Infomap, which was proposed by [30]. This algorithm uses a new information theory to detect communities in a complex network. The last one is the GA-Net algorithm. It was proposed by

Pizzuti [27]. This algorithm uses a single-objective genetic algorithm to optimize a fitness function, called community score (CS), for detecting communities inside a network.

5.1 Experimental settings

We chose two evaluation criteria to evaluate our proposed method: the modularity (Q), and the Normalized Mutual Information (NMI).

The normalized mutation information (NMI) measures the similarity between two communities, the truly known community of the network and the detected one. Suppose that A and B are two partitions of a network and C is the confusion matrix, where C_{ij} is the number of nodes that are in both communities i and j of partitions A and B respectively. Then $NMI(A, B)$ is calculated as follows:

$$NMI(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log(C_{ij}N / C_i.C_j)}{\sum_{i=1}^{C_A} C_i \log(C_i/N) + \sum_{j=1}^{C_B} C_j \log(C_j/N)} \quad (15)$$

Where $C_A(C_B)$ is the number of communities in partition $A(B)$, $C_i(C_j)$ is the sum of elements of C in row i (column j), and N is the number of nodes of the network. If $A = B$, then $NMI(A, B) = 1$; if A and B are totally different, then $NMI(A, B) = 0$.

5.2 Experiments on GN extended benchmark networks

We first evaluated our proposed method on the GN extended benchmark networks to see its effectiveness against other methods. The GN extended benchmark networks proposed by Lancichinetti et al. [17] is an extension of the classical GN benchmark network proposed by Girvan and Newman in [24]. It consists of 128 nodes divided into four communities with equal size of 32 nodes. The average degree of each node is 16. μ is a mixing parameter, used to control the percentage of node links between the nodes of its community and the total nodes of the network. When $\mu < 0.5$, the network has a strong community structure. On the contrary, the community structure is ambiguous, and the detection of its structure will be difficult.

We have compared our proposed method (DMFWA) against Infomap, CNM, and GA-Net algorithms on 10 GN extended networks by ranging the μ mixing parameter value from 0.05 to 0.5. Results of these experiments are given in Figs. 5 and 6.

Figure 5 shows the average NMI values obtained from different algorithms over 30 runs, with μ mixing parameter ranging from 0.05 to 0.5. We can observe from this figure that when $\mu \leq 0.2$, the proposed algorithm and the three other algorithms can successfully detect the exact community structure of the network. When $\mu > 0.2$, the network community structure becomes more and more ambiguous.

Infomap, CNM, and GA-Net algorithms rapidly became unsuccessful and failed to detect the exact network community structure. Infomap first shows its weakness, and its detection capacity decreases dramatically from $\mu \geq 0.2$. CNM and GA-Net algorithms detection capacity fall from $\mu \geq 0.25$ and $\mu \geq 0.3$ respectively. On the contrary, our proposed algorithm shows its dominance over the compared algorithms, and its detection capacity is stable.

Obtained average modularity (Q) values from different algorithms are shown in Fig. 6. From reported values in this figure, we can observe that when $\mu \leq 0.15$. All algorithms, detect with great success the community structure of every generated network. However, starting from $\mu = 0.2$, the detection task becomes difficult, and the detection capacity of each algorithm is decreased. CNM algorithm gave the worst detection result among the other algorithms, when $\mu = 0.2$. Obtained modularity value from this algorithm is equal to 0.5, which is smaller than other values obtained from, GA-Net (0.5201), Infomap (0.5598), and our method (0.5982). This decrease in modularity values continues considerably, and reaches the max at $\mu = 0.35$. At this value of μ , the community structure of the generated network is very ambiguous. Infomap, CNM, and GA-Net algorithms fail to detect the communities. At the same time, our proposed algorithm succeed in detecting them, with a value of modularity equal to 0.3236, greater than the other values, including: Infomap (0.2063), GA-Net (0.1659), and CNM (0.1336).

These reported modularity values in Fig. 6, clearly exhibit the advantage of our proposed algorithm against the other compared algorithms.

Table 1 shows the statistical modularity values obtained over 30 individual runs of each algorithm on the GN extended benchmark networks with different μ values. A non-parametric statistical test using the Wilcoxon's rank sum test [9, 37] has also been performed at 5 % (0.05) significance level on the modularity (Q) data. As a null hypothesis (H_0), we assumed that there is no significant difference between the mean values of the two observations sets (modularity values). While the alternative hypothesis (H_1) is that there is a significant difference in the average values of the two sets. When the p-value is less than 0.05, the null hypothesis will be rejected. This indicates that our method performs statistically more efficiently than the compared algorithm.

All the p-values recorded in Table 1 are smaller than 0.05, this is against the null hypothesis. This means that the null hypothesis will be rejected, and the average modularity values obtained by our proposed method are statistically significant. From these p-values, we conclude that our proposed method outperforms the three other algorithms on the GN extended benchmark networks. Except for the first experiment where the p-values are close enough to 1,

Fig. 5 Average NMI values over 30 runs on GN extended benchmark networks

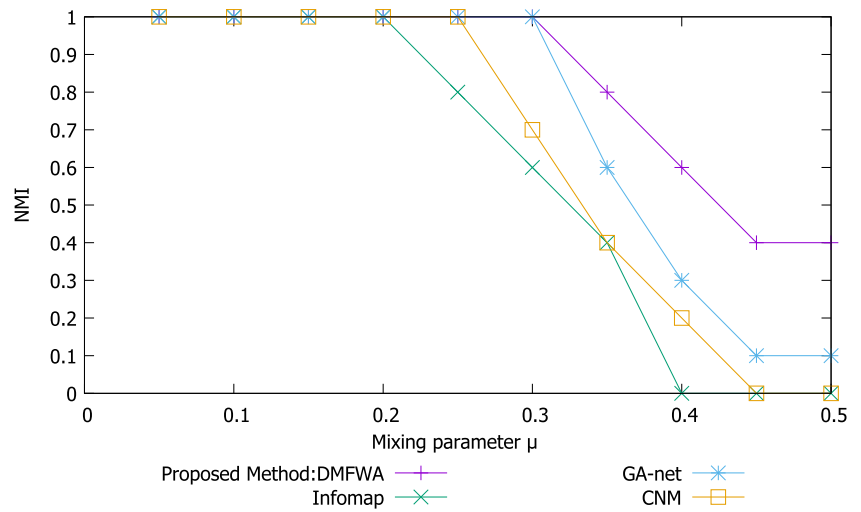


Fig. 6 Average Q values over 30 runs on GN extended benchmark networks

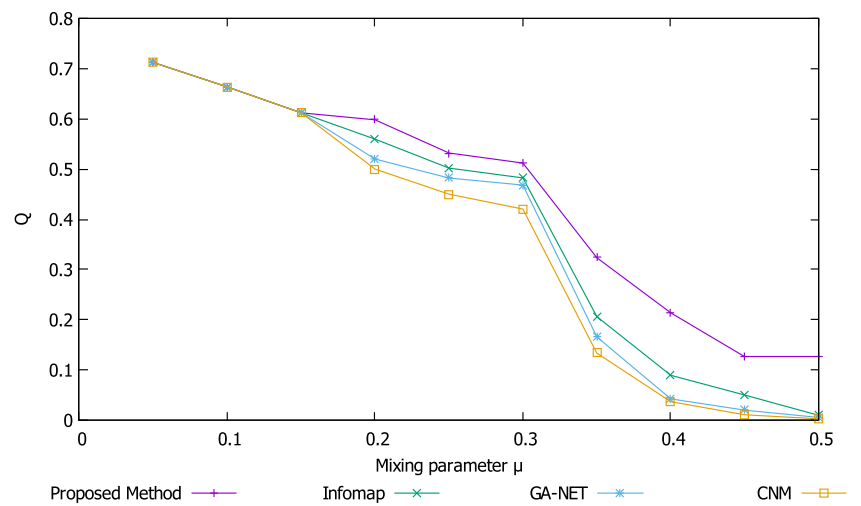
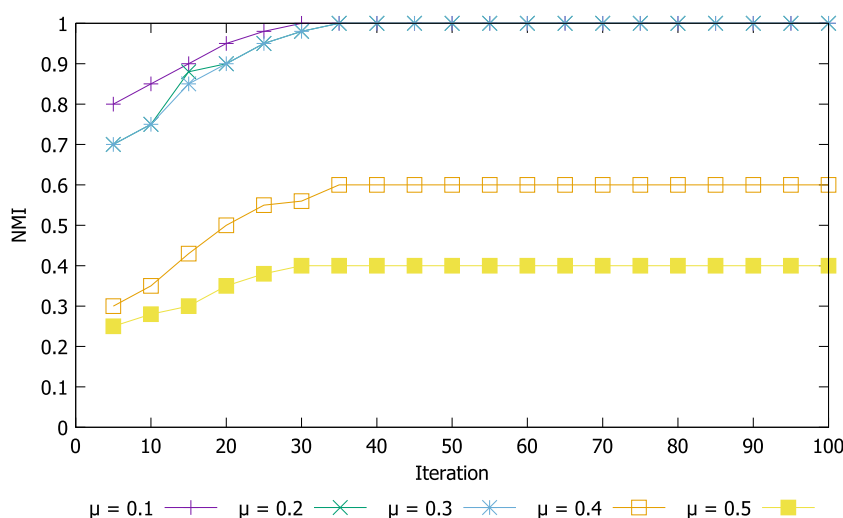


Table 1 Statistical results obtained over 30 runs on GN extended benchmark networks. The “p-value” is produced by Wilcoxon’s rank sum test taking comparing our proposed method with other algorithms

Network	Index	Proposed Algorithm	Infomap	GA-net	CNM
GN ($\mu = 0.1$)	Q_{avg}	0.6638	0.6638	0.6638	0.6638
	p-value	×	0.8493	0.8483	0.8653
GN ($\mu = 0.2$)	Q_{avg}	0.5982	0.5598	0.5201	0.5000
	p-value	×	0.0388	0.0254	0.0198
GN ($\mu = 0.3$)	Q_{avg}	0.5121	0.4825	0.4683	0.4205
	p-value	×	0.0238	0.0191	0.0138
GN ($\mu = 0.4$)	Q_{avg}	0.2138	0.0895	0.0425	0.0369
	p-value	×	0.0205	0.0175	0.0126
GN ($\mu = 0.5$)	Q_{avg}	0.1265	0.0098	0.0056	0.0028
	p-value	×	0.0385	0.0162	0.0119

Fig. 7 Convergence of our proposed algorithm



which means that there is no significant difference between the modularity values obtained by the four algorithms. The average Q values reported in the same Table prove that.

To examine the convergence of our proposed algorithm, we evaluated it on the GN extended benchmark networks with mixing parameter $\mu \in [0.1, 0.2, 0.3, 0.4, 0.5]$ over a number of iterations ranging from 10 to 100. Results from this experiment are shown in Fig. 7. As is shown in this figure, we can observe that our proposed algorithm converges rapidly to a stable value of NMI, regardless the value of μ , starting from iteration 30. This is due to the impact of the initialization strategy and the mutation operator proposed in this paper that make the algorithm faster in converging to an optimal solution. One conclusion that can be drawn from results shown in Figs. 5, 6, and 7, is that our proposed algorithm considerably surpasses all other compared algorithms and has a high capacity of detecting the network community structure.

We presented in this section the experimental results obtained from our proposed method and three other well-known algorithms on the GN extended benchmark networks. However, this synthetic benchmark shows some limitations, such as the size of each community, which is the same for all communities. So some important features of real networks cannot be reflected. Due to this, we have performed another experiments on real-world

networks. Results from these experiments will be presented in the next section.

5.3 Experiments on real-world networks

We present now, the results from evaluating our proposed algorithm (DMFWA) on four real-world networks with known community structures including Zachary’s karate club network [38], Dolphins social network [22], Krebs’ Books on US Politics network ¹, and American College Football network [10]. These networks are described as follows:

1. *Zachary’s Karate Club* This is one of the most popular networks that has been widely used in literature. This network represents a social network of a karate club members studied by [38]. It consists of 34 nodes, with each node representing a member of this club. However, during this study a dispute occurred between the administrator (node 34) and the instructor (node 1), which led to separate the club members into two smaller club groups. One group, formed around the administrator, contains 16 members. The other 18 members formed a group around the instructor.
2. *Dolphins social network* This social network, represents results from studying the behavior of 62 bottlenose dolphins among seven years in Doubtful Sound, New Zealand [23].
3. *Krebs’ Books* Krebs’ Books on politics network contains 105 US politics books, collected by Krebs and that have been sold by Amazon.com.
4. *American College Football* American College Football network [10], which consists of 115 US Football teams

Table 2 Characteristics of used real-world networks

Network	Nodes	Edges	Real communities
Karate	34	78	2
Dolphin	62	159	2
Politics	105	613	3
Football	115	616	12

¹Books about US Politics: <http://networkdata.ics.uci.edu/data.php?d=polbooks>

Table 3 Experiment result on real-world networks with comparison to Infomap, CNM, and GA-Net algorithms

Network	Evaluation criterion	Proposed Algorithm	Infomap	CNM	GA-net
Karate	NMI_{max}	1	0.7851	0.7691	0.6921
	NMI_{avg}	1	0.7672	0.7603	0.6805
	Q_{max}	0.4198	0.4091	0.4084	0.4103
	Q_{avg}	0.4038	0.3991	0.3989	0.4022
	p-value	×	0.0388	0.0358	0.0392
Dolphin	NMI_{max}	0.9650	0.7049	0.6980	0.8069
	NMI_{avg}	0.9542	0.705	0.6873	0.8005
	Q_{max}	0.5093	0.4312	0.4569	0.4743
	Q_{avg}	0.5013	0.4308	0.4498	0.4704
	p-value	×	0.0328	0.0348	0.0382
Politics	NMI_{max}	0.7981	0.5218	0.5301	0.6809
	NMI_{avg}	0.7971	0.5125	0.5298	0.6795
	Q_{max}	0.5271	0.4265	0.4298	0.5816
	Q_{avg}	0.5081	0.4201	0.4193	0.5803
	p-value	×	0.0326	0.0342	0.0674
Football	NMI_{max}	0.9230	0.6853	0.7695	0.9120
	NMI_{avg}	0.9205	0.6845	0.7687	0.9105
	Q_{max}	0.6046	0.3494	0.5386	0.6012
	Q_{avg}	0.6041	0.5485	0.5381	0.6007
	p-value	×	0.0219	0.0291	0.0315

from 12 different regions of the USA, playing a championship game between each other during the season of fall 2000. Each link represents a game between two teams, a total 616 games have been played.

Table 2 gives characteristics of each network.

Results from these evaluations are shown in Table 3 along with the p-values obtained by the Wilcoxon’s rank sum test at 5 % (0.05) significance level on the modularity (Q) data. We can observe from this table that our proposed algorithm performs well on these real-world networks and outperforms other compared algorithms. However, the

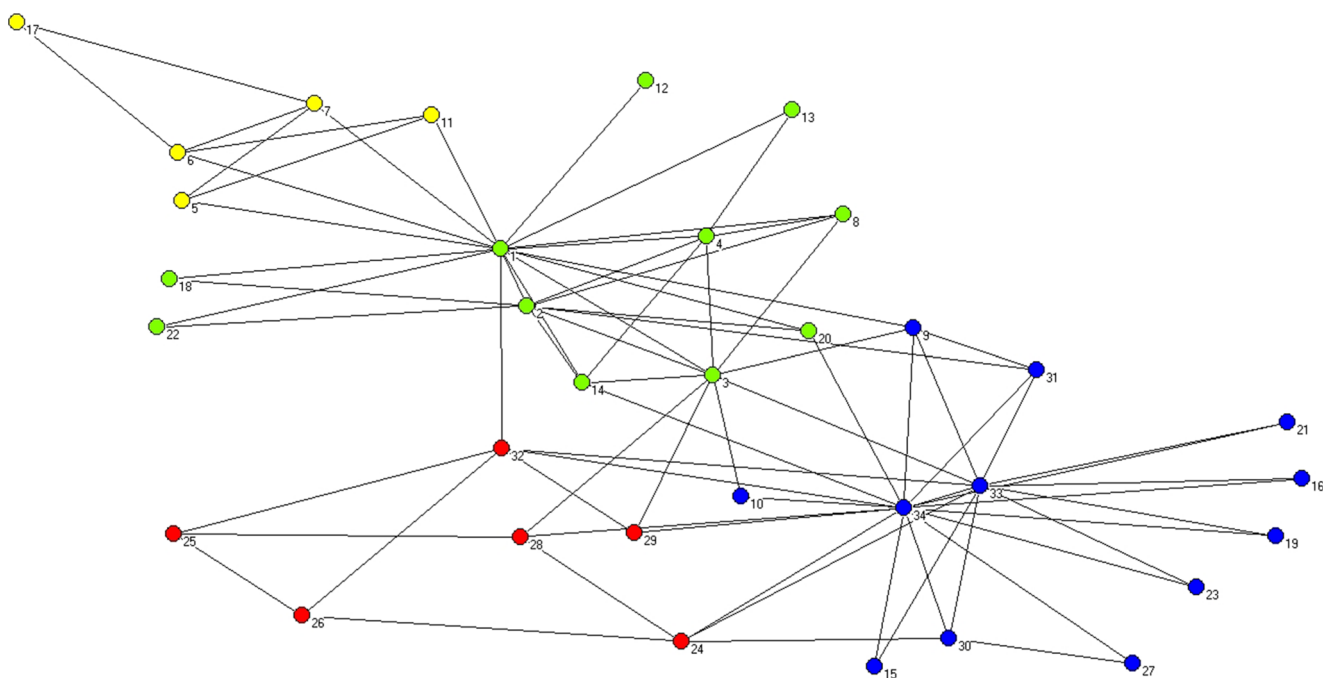


Fig. 8 Community structure of Zachary’s Karate Club network obtained by our proposed algorithm

GA-Net algorithm shows a better performance over our proposed algorithm on the Krebs' Books on US Politics network, with an average value of modularity equal to 0.5803, and 0.5081 for our algorithm. This small difference in value is due to the fact, that the CNM algorithm optimizes an efficient fitness function, called community score, to detect communities inside a network which has worked well on that network.

Always from Table 3, we can observe that there is an interesting difference between NMI values obtained from our proposed algorithm, and the three other ones. This difference is clearly seen on the Karate club network, when our proposed algorithm successfully detects the correct community structure. The obtained NMI value on this network is equal to 1 for our algorithm, 0.7851 for Infomap, 0.7691 for CNM, and 0.6921 for GA-Net algorithm. From obtained NMI values on the Krebs' Books on US Politics network, we can observe that our algorithm gives the best NMI value, equal to 0.7981. Which means that our proposed algorithm, succeed in detecting a community structure that is close enough to the correct one.

From the p-values in Table 3, which are all less than 0.05, we conclude that our proposed method performs statistically more efficiently than the other three algorithms on all the presented networks, except for the Politics Books network, where the GA-Net algorithm shows better results with a p-value equal to 0.0674.

Figure 8 shows the community structure of the Zachary's Karate Club network obtained by our algorithm. As is shown in this figure, the network is divided into four communities with a value of modularity Q equal to 0.4198. Nodes with blue color represent members who have joined the group of the administrator of the club (node 34). While nodes with green color are members who joined the group of the instructor (node 1). Node with red and yellow colors, are members who chose to stay away from the dispute.

6 Conclusion

The Fireworks Algorithm (FWA) is a new evolutionary swarm intelligence algorithm which has been widely used in many fields as a continuous optimization algorithm. However, its applications to discrete problems are almost missing. In this paper, we present a new discrete modified variant of the fireworks algorithm to solve the problem of community detection in complex networks by maximizing the modularity density value. Our main contributions are on the design of a new initialization and mutation strategies, based on the label propagation method.

Results from evaluating our proposed algorithm on real-world and synthetic networks compared to three other

algorithms, show the superiority of our proposed algorithm over the others.

As future works, the proposed algorithm in this paper uses the modularity density as a single-objective function. However, it can be extended to a multi-objective optimization technique. We also plan to test our proposed algorithm on signed networks since the experiments presented in this paper are all done on unsigned networks.

References

- Brandes U, Delling D, Gaertler M, Görke R, Hofer M, Nikoloski Z, Wagner D (2006) Maximizing modularity is hard. arXiv preprint physics/0608255
- Cai Q, Ma L, Gong M, Tian D (2014) A survey on network community detection based on evolutionary computation. *Int J Bio-Inspired Comput*
- Cai Q, Gong M, Ma L, Jiao L (2015a) A novel clonal selection algorithm for community detection in complex networks. *Comput Intell* 31(3):442–464
- Cai Q, Gong M, Ma L, Ruan S, Yuan F, Jiao L (2015b) Greedy discrete particle swarm optimization for large-scale social network clustering. *Inf Sci* 316:503–516
- Clauset A, Newman ME, Moore C (2004) Finding community structure in very large networks. *Phys Rev E* 70(6):066,111
- Ding K, Zheng S, Tan Y (2013) A gpu-based parallel fireworks algorithm for optimization. In: Proceedings of the 15th annual conference on Genetic and evolutionary computation, ACM, pp 9–16
- Fortunato S (2010) Community detection in graphs. *Phys Rep* 486(3):75–174
- Fortunato S, Barthelemy M (2007) Resolution limit in community detection. *Proc Natl Acad Sci* 104(1):36–41
- García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the cec2005 special session on real parameter optimization. *J Heuristics* 15(6):617–644
- Girvan M, Newman ME (2002) Community structure in social and biological networks. *Proc Natl Acad Sci* 99(12):7821–7826
- Gog A, Dumitrescu D, Hirsbrunner B (2007) Community detection in complex networks using collaborative evolutionary algorithms. In: *Advances in Artificial Life*. Springer, pp 886–894
- Gong M, Fu B, Jiao L, Du H (2011) Memetic algorithm for community detection in networks. *Phys Rev E* 84(5):056,101
- Gong M, Cai Q, Li Y, Ma J (2012) An improved memetic algorithm for community detection in complex networks. In: *Evolutionary computation (CEC), 2012 IEEE Congress on IEEE*, pp 1–8
- Gong M, Cai Q, Chen X, Ma L (2014) Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Trans Evol Comput* 18(1):82–97
- Guimera R, Sales-Pardo M, Amaral LAN (2004) Modularity from fluctuations in random graphs and complex networks. *Phys Rev E* 70(2):025,101
- Guoqiang C, Xiaofang G (2010) A genetic algorithm based on modularity density for detecting community structure in complex networks. In: *Computational intelligence and security (CIS), 2010 International Conference on IEEE*, pp 151–154
- Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys Rev E* 78(4):046,110

18. Li J, Zheng S, Tan Y (2014) Adaptive fireworks algorithm. In: *Evolutionary computation (CEC), 2014 IEEE Congress on IEEE*, pp 3214–3221
19. Li Z, Zhang S, Wang RS, Zhang XS, Chen L (2008) Quantitative function for community detection. *Phys Rev E* 77(3):036,109
20. Lipczak M, Milius E (2009) Agglomerative genetic algorithm for clustering in social networks. In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM*, pp 1243–1250
21. Liu X, Li D, Wang S, Tao Z (2007) Effective algorithm for detecting community structure in complex networks based on ga and clustering. In: *Computational Science–ICCS 2007*. Springer, pp 657–664
22. Lusseau D (2003) The emergent properties of a dolphin social network. *Proc R Soc Lond B Biol Sci* 270(Suppl 2):S186–S188
23. Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behav Ecol Sociobiol* 54(4):396–405
24. Newman ME (2006) Modularity and community structure in networks. *Proc Natl Acad Sci* 103(23):8577–8582
25. Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. *Phys Rev E* 69(2):026,113
26. Pang-Ning T, Steinbach M, Kumar V et al. (2006) Introduction to data mining. In: *Library of congress*, vol 74
27. Pizzuti C (2008) Ga-net: A genetic algorithm for community detection in social networks. In: *Parallel Problem Solving from Nature–PPSN X*. Springer, pp 1081–1090
28. Pizzuti C (2012) A multiobjective genetic algorithm to find communities in complex networks. *IEEE Trans Evol Comput* 16(3):418–430
29. Rahmani A, Amine A, Hamou RM, Rahmani ME, Bouarara HA (2015) Privacy preserving through fireworks algorithm based model for image perturbation in big data. *International Journal of Swarm Intelligence Research (IJSIR)* 6(3):41–58
30. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. *Proc Natl Acad Sci* 105(4):1118–1123
31. Shi Z, Liu Y, Liang J (2009) Pso-based community detection in complex networks. In: *Knowledge acquisition and modeling, 2009. KAM'09. Second international symposium on, IEEE*, vol 3, pp 114–119
32. Singhal A (2001) Modern information retrieval: a brief overview. *IEEE Data Eng Bull* 24(4):35–43
33. Suaris PR, Kedem G (1988) An algorithm for quadrisection and its application to standard cell placement. *IEEE Trans Circuits Syst* 35(3):294–303
34. Tan Y (2015) Discrete firework algorithm for combinatorial optimization problem. In: *Fireworks Algorithm*. Springer, pp 209–226
35. Tan Y, Zhu Y (2010) Fireworks algorithm for optimization. In: *Advances in Swarm Intelligence*. Springer, pp 355–364
36. Tasgin M, Herdagdelen A, Bingol H (2007) Community detection in complex networks using genetic algorithms. *arXiv preprint arXiv:07110491*
37. Wilcoxon F (1945) Individual comparisons by ranking methods. *Biom Bull* 1(6):80–83
38. Zachary WW (1977) An information flow model for conflict and fission in small groups. *J Anthropol Res*:452–473
39. Zhang B, Zhang M, Zheng YJ (2014) Improving enhanced fireworks algorithm with new gaussian explosion and population selection strategies. In: *Advances in Swarm Intelligence*. Springer, pp 53–63
40. Zheng S., Janecek A., Tan Y (2013a) Enhanced fireworks algorithm. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on, IEEE*, pp 2069–2077
41. Zheng S, Janecek A, Li J, Tan Y (2014) Dynamic search in fireworks algorithm. In: *Evolutionary computation (CEC), 2014 IEEE Congress on, IEEE*, pp 3222–3229
42. Zheng YJ, Song Q, Chen SY (2013b) Multiobjective fireworks optimization for variable-rate fertilization in oil crop production. *Appl Soft Comput* 13(11):4253–4263. doi:10.1016/j.asoc.2013.07.004. <http://www.sciencedirect.com/science/article/pii/S1568494613002305>
43. Zheng YJ, Xu XL, Ling HF, Chen SY (2015) A hybrid fireworks optimization method with differential evolution operators. *Neurocomputing* 148:75 – 82. doi:10.1016/j.neucom.2012.08.075. <http://www.sciencedirect.com/science/article/pii/S092523121400931X>



Mohamed Guendouz received his Bachelor's degree in Computer Science in 2012 and his Master's degree in 2014 from Tahar Moulay University of Saida, Algeria. In November 2014, he joined GeCoDe laboratory as a PhD student and a researcher. His research interests include: Big Data, Social Networks Analysis, Data Mining, and Evolutionary Computing. Mohamed has published a number of scientific journal articles and conference papers in the field of social networks analysis.

Abdelmalek Amine received an engineering degree in Computer Science from the Computer Science department of Djillali Liabes University of Sidi-Belabbes-Algeria, received the Magister diploma in Computational Science and PhD from Djillali Liabes University in collaboration with Joseph Fourier University of Grenoble. His research interests include Big data, data mining, text mining, ontology, classification, clustering, neural networks, and biomimetic optimization methods. He participates in the program committees of several international conferences and on the editorial boards of international journals. Prof. Amine is the head of GeCoDe-knowledge management and complex data-laboratory at UTM University of Saida, Algeria; he also collaborates with the “knowledge base and database” team of TIMC laboratory at Joseph Fourier University of Grenoble.



Reda Mohamed Hamou received an engineering degree in Computer Science from the Computer Science department of Djillali Liabes University of Sidi Belabbes, Algeria, and a PhD (Artificial Intelligence) from the same university. He has several publications in the field of BioInspired and Metaheuristics. His research interests include Data Mining, Text Mining, Classification, Clustering, Computational Intelligence, Neural Networks, Evolutionary Com-

putation and Biomimetic Optimization methods. Currently, Dr. Hamou is a head of a research team in GeCoDe laboratory. He is also an associate professor in the faculty of technology of Tahar Moulay University of Saida, Algeria.