Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

An Improved Grey Wolves Optimization Algorithm for Dynamic Community Detection and Data Clustering

Fatemeh Besharatnia, Alireza Talebpour & Sadegh Aliakbary

To cite this article: Fatemeh Besharatnia, Alireza Talebpour & Sadegh Aliakbary (2022) An Improved Grey Wolves Optimization Algorithm for Dynamic Community Detection and Data Clustering, Applied Artificial Intelligence, 36:1, 2012000, DOI: 10.1080/08839514.2021.2012000

To link to this article: https://doi.org/10.1080/08839514.2021.2012000

Published online: 31 Dec 2021.

APPLIED ARTIFICIAL INTELLIGENCE
2022, VOL. 36, NO. 1, e2012000 (948 pages)
https://doi.org/10.1080/08839514.2021.2012000

Taylor & Francis
Taylor & Francis Group

An Improved Grey Wolves Optimization Algorithm for Dynamic Community Detection and Data Clustering

Fatemeh Besharatnia, Alireza Talebpour, and Sadegh Aliakbary

Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran

ABSTRACT
One of the salient features of real-world networks such as social networks is the existence of community structures. Because of the importance of groups and communities in social networks, various algorithms have been proposed to identify communities in this type of dynamic networks. In this paper, we present a new approach to community recognition in dynamic social networks, which is multi-objective and metaheuristic. Our approach is to improve the Grey Wolf Optimizer algorithm and the Label Propagation algorithm and to combine the two algorithms for better performance. We performed our experiments on two artificial and real datasets, and the results show that our proposed method performs better compared to present algorithms in terms of both quality and detection speed. We also applied our proposed algorithm to 23 base functions, which performed better than the other metaheuristic algorithms. At the end, the performance of our proposed algorithm is compared to six other clustering methods on nine datasets from the UCI machine learning laboratory. The simulation results show the effectiveness of the proposed algorithm for solving data clustering problems.

ARTICLE HISTORY
Received 29 September 2020
Revised 17 September 2021
Accepted 23 November 2021

Introduction

Given that the community recognition has an optimization problem with high time complexity, the use of mathematical methods in large networks is very time consuming. Therefore, researchers use meta-heuristic methods such as genetic algorithms to solve this problem (Pizzuti 2012). The main problem with such meta-heuristic methods is getting trapped in the local optimal response. An optimization algorithm that has just been introduced is the Grey Wolf optimizer algorithm, which traverses the search space well and is less trapped in the local optimal answer trap than other algorithms (Mirjalili, Mirjalili, and Lewis 2014).

In this paper, we present the identification of communities in dynamic networks as a multi-objective optimization problem, first improving the grey wolf optimization algorithm (Mirjalili, Mirjalili, and Lewis 2014) by two approaches and then merging it with the improved label propagation algorithm, which caused our proposed algorithm to perform better for community detection.

CONTACT Alireza Talebpour talebpour@sbu.ac.ir Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, 1983969411, Iran

The problem of recognizing communities in a static network can be considered as a one-goal problem, but in dynamic networks, it is a multi-objective one, because in this type of network, two qualities of community must be examined. The first is the modularity criterion introduced by Girvan and Newman. This criterion calculates the quality of communities (Girvan and Newman 2002). The second is the temporal cost for difference among the present and prior time intervals, for which the NMI[1] criterion proposed by Dunn et al. is used (Dannon et al. 2005). We conducted our experiments on both artificial and real datasets, and our results show that our suggested algorithm performs very well compared to other algorithms. We also performed our proposed algorithm on 23 standard test functions, which are classical functions and minimal functions, which had better results than other meta-heuristic algorithms. Finally, the performances of the six other clustering methods on nine datasets from UCI[2] machine learning laboratory are compared to the presented algorithm (Blake and Merz 1998). In case of solving data clustering problems and according to the simulation results, the proposed algorithm is more effective. The average error of our proposed method, called IGWO-LP,[3] is 7.87% in the dataset, which is the lowest value among the baseline algorithms. The comprehensive evaluations confirm that the proposed algorithm outperforms baseline algorithms according to the average clustering error rate.

The main features of our work are as follows: our proposed algorithm is a combination of the improved multi-objective GWO algorithm and the improved LP[4] algorithm, which leads to better performance and higher accuracy of the proposed algorithm compared to other algorithms. Our proposed algorithm has linear time complexity. It was also performed on 23 basic functions, which has a better answer than most of other utilized baseline algorithms. Moreover, by comparing the proposed algorithm with six other clustering methods on nine datasets from UCI machine learning laboratory, it was shown that it is effective in solving data clustering problems.

The rest of the article is as follows: the second part is the research literature, which is a general overview of the related works in this field; in the third part, the proposed method is presented; and then in the fourth part, the results of the proposed algorithm with other algorithms are presented. Finally, the fifth part is the result of our work.

Literature

The issue of community detection has been widely studied in social networks due to its importance. As we know, this issue is generally divided into two main categories, static and dynamic; some of which related to static category are as follows

Moayedi Kia proposed a new multi-purpose community recognition algorithm by analyzing the importance of nodes. Their proposed method had two objective functions for identifying communities and their characteristics, and they estimated the importance of nodes using potential topological context and degree of convergence. They found information about the characteristics of the nodes, so that they could find the power to distinguish the nodes (Moayedikia 2018). In another study, Belal et al. proposed a framework for identifying communities by finding the first structure of community with the help of an evolutionary algorithm and then merging communities to improve the final structure of the community that has a high modularity (Bilal et al. 2017).

Pourkazemi et al. presented an article on a multi-objective optimization algorithm based on the particle swarm optimization algorithm. This algorithm uses an anti-axis method to generate an initial overlap. This algorithm optimizes two target functions at the same time, indicating a partition of the network and the use of a mutation operator to investigate the problem at high dimensions. Moreover, their proposed algorithm performs better by increasing the size of the network (Pourkazemi and Keyvanpour 2017).

Lin et al. suggested an algorithm based on the Girvan and Newman algorithm, which uses a coefficient of edge clustering in lieu of the middle edge to identify communities (Lin et al. 2014). In the second category, which are dynamic community detection algorithms in dynamic social networks, the following research can be mentioned:

Sami et al. detected communities on social networks by a two-step approach based on global and local information. This approach examines global information in the first stage and uses a new algorithm in the second stage to better explore communities (Samie and Hamzeh 2016). Folino et al. also presented a multi-objective genetic algorithm called DYNMOGA for the problem of community recognition in dynamic networks, and since their algorithm was multi-objective, they used Pareto optimality theory to find the answers. DYNMOGA uses dimensionless sorting to balance the quality of temporal cost and instantaneous cost. The disadvantage of this algorithm is that it is not scalable, which causes it to take a long time to run on large networks (Folino and Pizzuti 2014).

Quick Community Adaptation (QCA) is a modularity-based method that focuses directly on the variations in the network structure, so it does not recalculate community structure from scratch at every time phase (Nguyen et al. 2014). Darts et al. studied the edge density in community detection and presented insight for implementing it with other community detection methods (Ronhovde, Peter, and Nussinov 2013). The density measure of real number of edges divided by the possible number of edges was presented by them, in which a community attracts nodes, which results in a specified edge density threshold to stay satisfied. Lin et al. provided a framework called

Facetnet that does not address the problems of the traditional method and used a strong integrated process to identify communities; their method also organizes the evolution of the structure of society (Lin et al. 2008).

Liu et al. proposed a new multi-objective evolutionary clustering algorithm for classifying nodes and their neighbors using a migration operator as well as other useful operators. After that, by the use of a genome matrix, they encoded the structure information of networks for extending the search space. Moreover, the modularity was optimized by this genome matrix (Liu. et al. 2020). Besharatnia, Talebpour, and Aliakbary (2020), in the paper, presented a new algorithm to detect communities in dynamic social networks. Their method can identify communities progressively based on a multi-objective metaheuristic algorithm using label propagation technique (Besharatnia, Talebpour, and Aliakbary 2020).

Proposed Approach

Problem Description

We have combined two algorithms to identify communities in dynamic social networks. One of these algorithms is the GWO algorithm, which is a metaheuristic algorithm, and we have improved it. The next algorithm is the improved label propagation algorithm, which we used in each repetition of the GWO algorithm. This proposed label propagation algorithm performs better because unlike the basic label propagation algorithm, it does not treat all nodes equally, so it prevents the wide spread of labels, which is one of the most important problems of the basic algorithm. We named our proposed algorithm Improved Grey Wolf Optimizer Label Propagation (IGWO-LP), which is a multi-objective algorithm. We also assumed that all nodes have the same values and that communities do not overlap.

Community Definition

There are different definitions for community because of its significance in present social networks; however, it can be defined as: “A subset of vertices, with high density between nodes inside the group but fair density between the groups.” Figure 1 shows a sample of community structure. Surely, meta communities can be formed by combining communities (Pourkazemi and Keyvanpour 2017).

Defining Dynamic Communities

A network snapshot was considered, which was modeled by a weighted graph $G_{\mathrm{t}} = (V_{\mathrm{t}}\ E_{\mathrm{t}})$, in which V_t is the collection of nodes and E_t is the collection of edges at time step t. A network that is changing during T time steps is defined

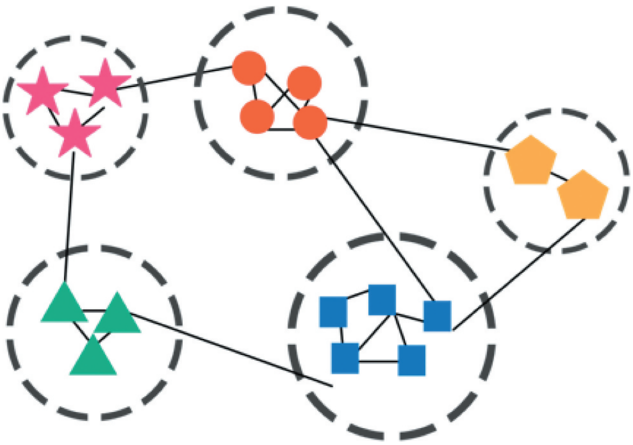

Figure 1. A sample of a network with five community structure.

as $G = \{G_1, G_2, \ldots, G_T\}$. The collection of k communities of network G_t at time step t can be defined as $C_t = \{C_{t1}, C_{t2}, \ldots, C_{tk}\}$, where $C_{tp} \cap C_{tq} = \varnothing$, $p, q \in \{1, 2, \ldots, k\}$ (Xinzheng, Weiyu, and Wu 2017).

Pareto Optimal Definition

As mentioned earlier, our problem is a multi-objective optimization problem. In this type of problem, there is no optimal answer, but there is always a set of answers that are good for the decision maker. One solution to this type of problem is to turn the problem into a goal using an equation (Chi et al. 2009).

$$\cos t = \alpha SC(G_i, C_i) + (1 - \mathrm{a})\mathrm{TC}(C_{i-1}, C_i) \tag{1}$$

Consider Equation (1) in which SC is the snapshot cost and TC is the temporal cost. To equilibrate the two functions, parameter α (0 ≤ α ≤ 1) is used. The algorithm discovers an optimal community using Equation (1). This method has a main limitation with respect to the value of α, which is necessary as an input parameter to equilibrate these two functions. Therefore, discovering the optimal value for the parameter α is a challenging issue in many cases, so we utilized Pareto optimality theory. For two partitions $C_1, C_2 \in \Omega$, partition C_1 dominates partition C_2 if and only if:

$$\begin{aligned} &\mathrm{i} \in \{1, \ldots, m\}\mathrm{P_i}(C_1)\mathrm{P_i}(C_2) \wedge \exists \mathrm{i} \in \{1, \ldots, m\}\mathrm{P_i}(C_1) <_{\mathrm{i}} (c_2) < tex \\ &='''' > /p\mathrm{i}(c_2) < > \end{aligned} \tag{2}$$

If there is no other partition dominating C, Partition $C \in \Omega$ is a Pareto optimal. Pareto optimal set is the collection of all Pareto optimal partitions and Pareto Front set is its related set under the objective function space (Pourkazemi and Keyvanpour 2017).

Proposed Method

The grey wolf algorithm is inspired by nature, which has been proven to have higher search accuracy and precision than similar algorithms such as the particle swarm optimization algorithm and the genetic algorithm (Mirjalili, Mirjalili, and Lewis 2014). In this section, an advanced version of the Grey Wolf Optimization Algorithm, which is suitable for detecting community on dynamic social networks and data clustering issues, is presented (Figure 1). This algorithm was developed by two methods. The improved GWO algorithm also explores the search space well and converges toward the best answer. However, it seems logical that the existing algorithms could be used to accelerate and improve this convergence operation for repairing the answers. In such a way that after each iteration of the improved GWO algorithm, the set of answers obtained is corrected by a correction algorithm. One of the popular algorithms in the field of community detection is the label propagation algorithm, which we improved based on the homogeneity criterion (Zhu and Ghahramani 2002). Algorithm and flowchart of the proposed method are seen in Figures 2–4.

```
Give initial values to a, A, C and number of population(n) and highest number of iteration Maxiter
Put t=1
For (i=1 ; i<=n ; i++)
    Create a primary population $X_i(t)$ by random
    Assess the fitness function of each wolf $f(X_i)$
End for
Set the values of best wolves $X_\alpha$, $X_\beta$, $X_\delta$ ,respectively
do{
    For (i=1 ; i<=n ; i++)
        Update every wolf in the population as in $\overrightarrow{x_{(t+1)}} = \frac{\overrightarrow{x_1'}+\overrightarrow{x_4}+\overrightarrow{x_5}}{3}$
        Reduce the parameter a from 2 to 0
        Update A, C
        Asses the fitness function of each wolf $f(X_i)$
    End for
    Update $X_\alpha$, $X_\beta$, $X_\delta$
    Applying Modified Label Propagation on $X_\alpha$, $X_\beta$ and $X_\delta$
    Set t= t+1
}While (t $\geq$ Maxiter)
Generate the best answer $X_\alpha$
```

Figure 2. Pseudo-code of IGWO-LP algorithm.

```
1. Give a unique initial label to all nodes
2. Calculate the degree of homogeneity between each pair of nodes
3. Generate a random visit list for all nodes
4. Update each node label to the neighbor label having highest amount of homogeneity
5. Repeat stage 4 until the label of each node is equal to the label of most of its neighbors
```

Figure 3. Pseudo-code of improved label propagation algorithm.

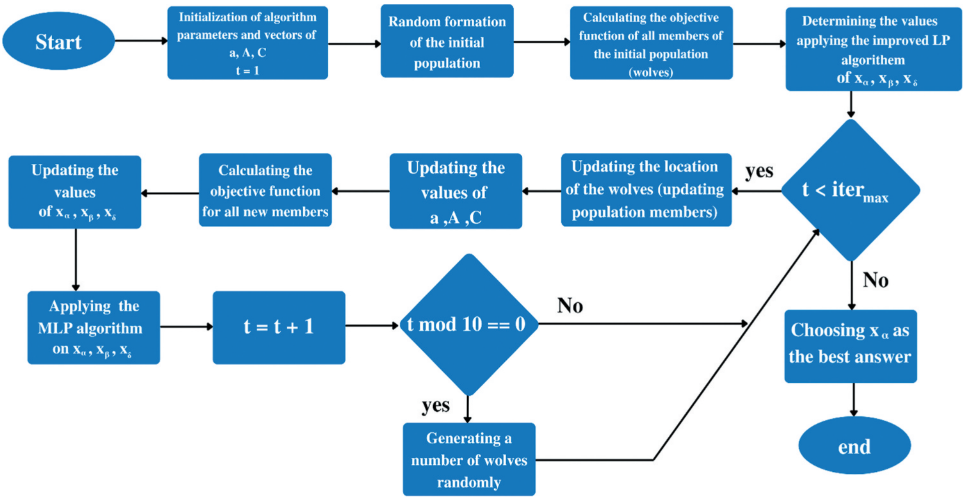
Start
Initialization of algorithm parameters and vectors of a, A, C t = 1
Random formation of the initial population
Calculating the objective function of all members of the initial population (wolves)
Determining the values applying the improved LP algorithem of x_α, x_β, x_δ
t < iter$_{max}$
yes
Updating the location of the wolves (updating population members)
Updating the values of a ,A ,C
Calculating the objective function for all new members
Updating the values of x_α, x_β, x_δ
Applying the MLP algorithm on x_α, x_β, x_δ
t = t + 1
t mod 10 == 0
No
yes
Generating a number of wolves randomly
No
Choosing x_α as the best answer
end

Figure 4. Flowchart of the IGWO-LP algorithm.

Coding and Decoding of Problem Answers by Grey Wolf Algorithm

Each member in the wolf population is coded as $X_p = \left\{x_p^1, x_p^2, \ldots, x_p^n\right\}$, where n is the number of network nodes and x_p^i represents the node i community label in the section X_p. In most cases, there exists n community in a network with n nodes, and the value x_p^i can be any integer between 1 and n. In the initialization step, X_p is assigned randomly. In the coding phase, nodes with equal community tags are grouped into a community. If there is k different community tags in a member at the end of the algorithm, the community structure corresponding to that member will automatically have k communities. A very important feature of this definition is the lack of need for prior knowledge of the value of k, which is a major problem in many community recognition algorithms. With this initial value, the grey wolf algorithm performs the optimization operation and reaches the best answer after convergence.

In nature, the number of wolves that work together to achieve greed is limited, but this optimization algorithm is mathematically defined in such a way that there is no limit to the number of wolves. Theoretically, the number of wolves can be any value, and more or less the number of wolves will not affect the operation of the algorithm. It is natural that a large number of wolves increase the quality of the algorithm and a small number of wolves increase the speed of the algorithm. Therefore, it is important to choose the right number of wolves, which will be done experimentally to find a relationship between the number of nodes and the right number of wolves.

First Technique

In the grey wolf algorithm, a random population of grey wolves is generated first to start the search process. The hunting process is normally driven by alpha, and during the iteration of the algorithm, the alpha, beta, and delta wolves estimate the probable position of the prey, and then all available answers (wolves) update their position according to the prey position. To obtain the mathematical model of the prey, we reduce the value of a. Note that the range of changes of "A" is reduced by "a." In other words, "A" is a random value in the range of [−2a, 2a] that "a" decreases from 2 to 0 under the iteration path. Parameter "a" is reduced to emphasize the local and global search between these two number ranges. Also, when | A | < 1, the wolves are forced to attack the prey, and when | A | > 1, the wolves move away from the prey. Local and global search is highly dependent on how the parameters work to update the population of candidate solutions. Finally, by this method, the alpha, beta, and delta values get closer and thus reduce local and global search to find the global optimal. Reducing local search will decrease the possibility of falling into the optimal local trap of the grey wolf algorithm. In this article, two techniques are used to improve local and global search (Zare, Hamidi, and Rahati 2015):

Local Search Technique. To improve the performance of grey wolves during prey attacks, a number of wolves in the β and δ groups are added to the alpha wolf class. (We take 10% of delta wolves and 5% of beta wolves and add them to alpha wolves. Since beta wolves are better than delta wolves, we add a smaller percentage of them to alpha wolves.) In fact, by increasing the effect of alpha responses and decreasing the effect of beta and delta responses, the prey attack performance improves. Equations (3–6) present the improved grey wolf algorithm method:

$$\overrightarrow{D_\alpha} = \left|C_1 . \overrightarrow{x_\alpha} - \vec{x}\right|, \overrightarrow{D_B} = \left|\overrightarrow{C_2} . \overrightarrow{x_B} - \vec{x}\right|, \overrightarrow{D_\delta} = \left|\overrightarrow{C_3} . \overrightarrow{x_\delta} - \vec{x}\right| \tag{3}$$

$$\overrightarrow{x_1} = \overrightarrow{x_\alpha} - \overrightarrow{A_1} . \left(\overrightarrow{D_\alpha}\right), \overrightarrow{x_2} = \overrightarrow{x_\alpha} - \overrightarrow{A_2} . \left(\overrightarrow{D_\alpha}\right),$$

$$\overrightarrow{x_3} = \overrightarrow{x_\alpha} - \overrightarrow{A_3} . \left(\overrightarrow{D_\alpha}\right), \overrightarrow{x_4} = \overrightarrow{x_B} - \overrightarrow{A_4} . (\overrightarrow{D_B})$$

$$\overrightarrow{x_5} = \overrightarrow{x_\delta} - \overrightarrow{A_5} . \left(\overrightarrow{D_\delta}\right) \tag{4}$$

$$\overrightarrow{x_1'} = \frac{\overrightarrow{x_1} + \overrightarrow{x_2} + \overrightarrow{x_3}}{3} \tag{5}$$

$$\overrightarrow{x_{(t+1)}} = \frac{x_1' + \overrightarrow{x_4} + \overrightarrow{x_5}}{3} \tag{6}$$

The Second Technique

As mentioned earlier, the grey wolf suffers from two problems: premature convergence and trapping in local optimal conditions.

Based on this, the theory of chaos or the theory of irregularities is used in this article. Chaos theory is used to study the dynamic systems of chaos. Chaos systems are nonlinear dynamic systems that are very sensitive to their primary conditions. A small change in the initial condition of such systems will cause many changes in the future. These systems include meta-heuristic algorithms that use the fortuity element. To create chaos in this proposed algorithm, after a certain number of repetitions of the algorithm, a number of wolves are randomly applied to the algorithm, so that if the algorithm deviates from its path, it can be directed to the correct and optimal path.

Moreover, to improve the response, after each iteration of the proposed algorithm, the set of responses obtained would be improved by the proposed label propagation algorithm, which is described below. The basic label propagation algorithm, which selects the label of each node based on the majority of the neighbors' labels, considers all neighbors to be the same. However, in reality, each node plays a different role based on its characteristics in the network. In the real world, nodes with great resemblance tend to be grouped together. With this in mind, community detection methods that use an appropriate homogeneity criterion can yield to better results in practice. The proposed homogeneity criterion is inspired by resource allocation (Katoh and Ibaraki 1998); at the time, node i dispatches a resource to node j and their shared neighbors operate as transmitters. So the similarity between nodes i and j could be expressed as the amount of source received by j. This homogeneity criterion in a network will be calculated as follows:

$$R_{ij} = \sum_{n \in CN} d(n) \tag{7}$$

where CN is the set of common neighbor nodes and $d(n)$ is the degree of node n.

The proposed method will select the labels based on the correlation of the nodes, in which the labels are published based on the degree of homogeneity. This method is different from the original label propagation algorithm because it does not encounter all neighbors equally. The propagation steps are as follows: first, all nodes are given a unique initial label. The degree of homogeneity between each pair of nodes is calculated. Then, a random visit list is generated for all nodes. Each node label is updated to the neighbor label that has the highest amount of homogeneity. This operation will be repeated until the label of each node is equal to the label of most of its neighbors.

The number of neighbors with high homogeneity is expected to be much lower than the degree of each node. Put it differently, the selections will be limited during the label update step. This prevents the widespread publication of a label, which is one of the biggest drawbacks of the original label propagation algorithm. For example, if two nodes have the same label and they are not highly similar to other neighbors, their label will not be published to any other node. In other words, propagation will have a better segmentation rate by the amount of similarity. Assuming that the improved algorithm will perform better in publishing the label, it can also be formulated and generalized for labeling subnets.

The most important challenge that is solved by the grey wolf optimization algorithm is to avoid falling into the local optimal response. In other metaheuristic optimization algorithms, all members of the population converge toward that member of the population who gives the best solution. For example, in the particle swarm optimization algorithm, all particles move toward a particle having the best value, and if the particle having the best value is close to the optimal local response, the rest of the particles converge to the same local response. However, in the GWO algorithm, in the initial iterations of the algorithm, part of the population explores the search space so that even if the best answers are close to the local optimal, there is a chance that other members will explore the search space independent of the best answers to find a better answer. In consecutive iterations, the algorithm changes from the problem space search phase as well as the convergence to the best answer to the convergence phase to the best answer to explore around the best answer obtained in the final iterations. Using the proposed label correction algorithm along with the improved GWO algorithm makes the search space even better by generating answers that will not be generated using the grey wolf algorithm. In the proposed algorithm, according to the two approaches mentioned, the IGWO-LP algorithm is less in the local optimal. Moreover, using the proposed label propagation algorithm with our algorithm makes it even better to explore the search space. It is done by generating answers that will not be generated using the GWO algorithm.

Results

Performance Appraisal of the IGWO-LP Algorithm for the Problem of Detecting Communities on Dynamic Social Networks

The performance evaluation of the IGWO-LP algorithm with the DYNMOGA (Folino and Pizzuti 2014), GLmetric (Samie and Hamzeh 2016), and FacetNet (Lin et al. 2008) algorithms and comparison of the results with various types of real and artificial data are presented in this section. MATLAB software is used to develop the IGWO-LP algorithm. The parameters of the algorithms are seen in Table 1 and our criterion for measuring algorithms are also NMI. This criterion was described in the next section.

Evaluation Criteria

As mentioned, the NMI criterion was used to evaluate our proposed algorithm compared to other comparable algorithms. The NMI criterion is used for the temporal cost, which shows the difference between the time intervals to explain the resemblance among the structures of the current and prior community. It is assumed that A and B are the two partitions of a network and C is the confusion matrix, in which C_{ij} indicates the number of nodes shared in common by community i in partition A and community j in partition B. So NMI can be obtained as following (Danon et al. 2005):

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log\left(C_{ij}N/C_iC_j\right)}{\sum_{i=1}^{C_A} C_i\log\left(\frac{C_i}{N}\right) + \sum_{j=1}^{C_B} C_j\log\left(\frac{C_j}{N}\right)} \tag{7}$$

where N is the number of nodes, C_A (C_B) is the number of communities in Section A(B) and C_i (C_j) is the summation of elements of C in row of i and column of j. The value of NMI is one, if A = B and it is zero if two partitions are totally diverse.

Dataset 1 (Artificial Network*)*

The two datasets, SYN-FIX and SYN-VAR, which are proposed by Girvan and Newman are used. In the SYN-FIX network, there are 128 nodes divided into 4 communities, each with 32 nodes. The average degree of every node is 16 and has z (3 and 5) edges, which are connected to nodes out of the community. We have 10 time intervals in this dataset and at each time interval and in each community, three nodes leave the community at random and join another community (Girvan and Newman 2002).

Moreover, the SYN-VAR dataset has 256 nodes that are divided into 4 communities, meaning that each community contains 64 nodes. Like SYN-FIX, this dataset has 10 time intervals. For the second interval, 8 nodes are chosen from each community, creating a new community with 32 nodes, which is repeated until the fifth time interval. After that, the nodes come back to their primary communities. Therefore, the number of communities for 10 time intervals is: 4,5,6,7,8,8,7,6,5,4, respectively. The mean degree of every node is equivalent to half of the community size. Moreover, 16 nodes are omitted at random at each time interval and 16 new nodes are added.

Figures 5–8 show the NMI benchmark results for the SYN-FIX and SYN-VAR datasets. In all the time intervals, the IGWO-LP algorithm has a higher NMI value than other algorithms, which shows that this algorithm has a more accurate structure compared to other algorithms.

Dataset 2 (Real Network*)*

Call phone call: This dataset is a phone call cases between members of the fictitious Paraiso movement throughout the 10 days of June 2006. (We have 10 time intervals in this dataset) Every node presents an exclusive mobile phone and each edge indicates a call between two mobile phones. The number of nodes in this collection is 400, and telephone calls have been reported for

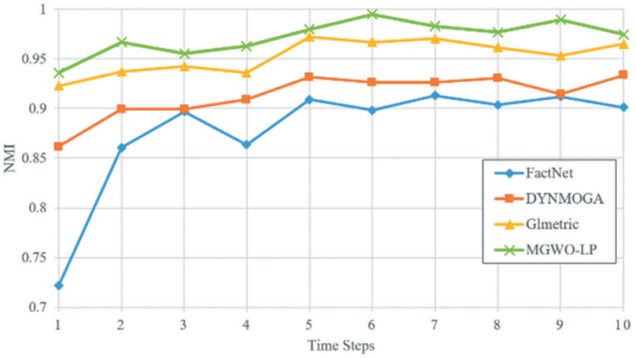

Figure 5. NMI results on SYN-FIX dataset (z = 3).

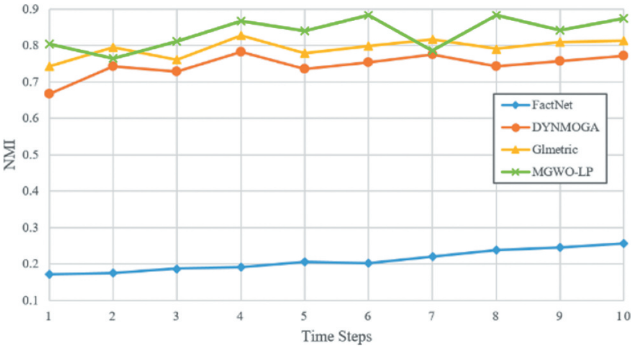

Figure 6. NMI results on SYN-FIX dataset (z = 5).

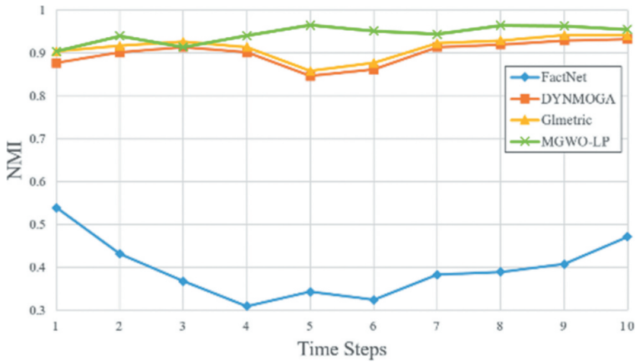

Figure 7. NMI results on SYN-VAR dataset (z = 3).

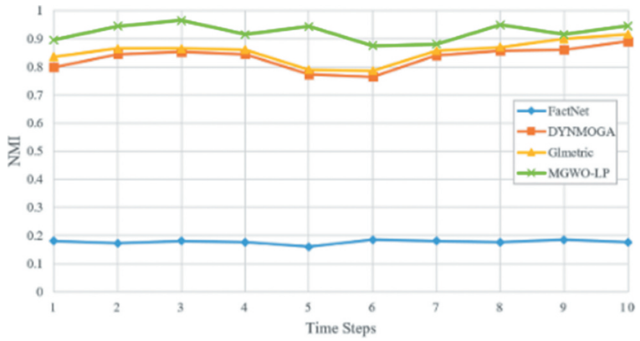

Figure 8. NMI results on SYN-VAR dataset (z = 5).

each day and time. The 5 major nodes in this network are: Ferdinando Catalano (node 201) and his brother Estaban (node 6), David Vidro (node 2) and his brothers Jorge and Juan (nodes 3 and 4). The mentioned 5 nodes altered their mobile phone number during the days 7 and 8. As a result, for the last three time intervals, their node numbers have changed from 201, 6, 2, 3, 4 to 301, 307, 310, 361 and 398, respectively. See Folino and Pizzuti (2014) to get further information regarding this dataset.

Since the actual structure of community is not identified, we followed the same approach used by Lin et al. (2009). We first applied the IGWO-LP algorithm to the whole network and considered the structure of the community obtained by the proposed algorithm as the structure of the real community, and then we calculated the Normalized Mutual Information (NMI) by the comparative algorithms on the network. As it is shown in Figure 9, the IGWO-LP is higher in all time intervals, indicating that this algorithm can build an accurate community structure in dynamic networks.

Dataset 3 (Real Network)

Enron mail dataset: (http://www.cs.cmu.edu/~enron/) this dataset includes email communication information from 1999 to 2002 from an American company. Its raw dataset includes 517,431 e-mails, including 151 customers and 3,500 folders, which we reduced the dataset to 50,000 after making the necessary corrections according to Lin et al. (2009). According to the 2001 data classification, 12 sub-categories of information are obtained based on the Gregorian calendar, and the number of communities in the network is 6. We used a similar method in the mobile dataset. Figure 10 shows the results of Normalized Mutual Information (NMI).

Dataset 4 (Real Network)

The football dataset is related to university information in the United States. This dataset was selected from 2005 to 2007 to compare algorithms. Here, each node identifies a team, and the edges indicate normal seasonal matches. In some communities, teams are divided into groups, and each team has to have match with members of the its community. There are 12 forums from 120 teams. Moreover, during these 3 years, the NMI values are between 0.6 and 0.7, which indicates the slow alterations of this dataset (Folino and Pizzuti 2010). Figure 11 shows the results of the NMI criterion of this dataset.

Comparing Methods

The results of the NMI criterion for all data sets can be seen in Table 2 and also Figure 12 shows the comparison between NMI values more clearly. And in terms of the NMI criterion, which is a maximal problem, the results show that for all of the above datasets in all of the above modes, the IGWO-LP algorithm performs better than other algorithms. These results indicate the superiority of the proposed algorithm compared to other algorithms. And using this criterion, we can say that our proposed algorithm has been more efficient in identifying the real structure of society.

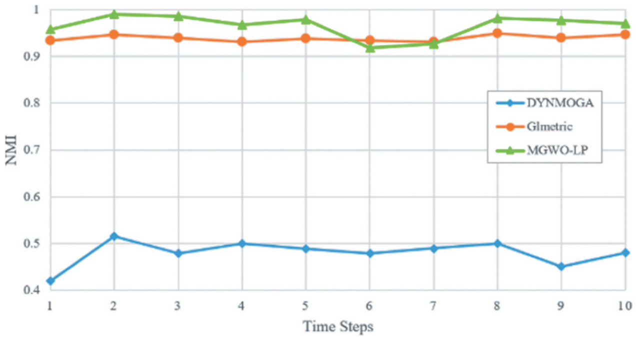

Figure 9. The NMI result of cell phone calls dataset.

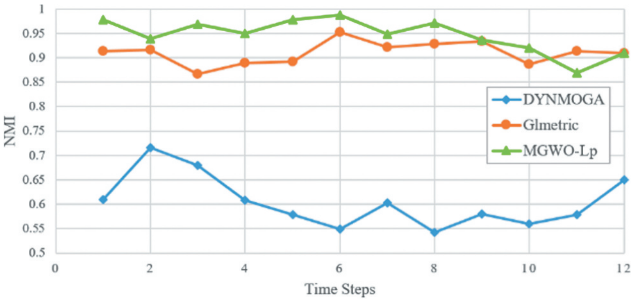

Figure 10. The NMI result of enron mail dataset.

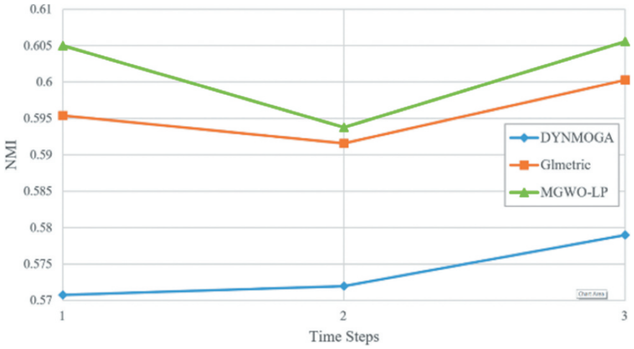

Figure 11. The NMI result of football dataset.

The performance of IGWO-LP algorithm for SYN-FIX dataset, when Z = 3, is better than other algorithms in all snapshots, and in SYN-FIX dataset when Z = 5, our proposed algorithm performed better than other algorithms in all snapshots except snapshots 7 and 2. In the SYN-VAR dataset, when Z = 3 and Z = 5, it can be said that the IGWO-LP algorithm performed better than the other algorithms.

In the cell phone calls dataset, our proposed algorithm performed worse than the GLmetric algorithm only in snapshots 6 and 7, and in the rest of the snapshots, our algorithm performed better than both DYNMOGA and Glmetric. Also, in case of the Enron mail data, IGWO-LP algorithm performed better in most snapshots than DYNMOGA and GLmetric algorithms, except for snapshots 11 and 12 which did not perform well compared to GLmetric.

Big – O Analysis

The time complexity of the IGWO-LP algorithm can be expressed in such a way that the time complexity for calculating the label propagation algorithm is O(m), where m is the number of edges in the graph, and the best solution in

Table 1. The parameters of the algorithm.

Parameter	Values
Run	30
Population number (number of wolves)	100
Number of generations (number of iterations)	100
Mutation rate	0.8
Elite reproduction	10%

the grey wolf optimization algorithm is O(p*m)+O(p) and the update of the wolves is O(p*n), where n is the number of nodes and p is the number of wolves. Time complexity for calculating modularity O(m) and NMI is also O(n) [25]. In addition, I is the number of repetitions of the algorithm and T is the number of intervals. Finally, the time complexity of the IGWO-LP algorithm is equal to:

$$\mathrm{IT}(\mathrm{pm} + \mathrm{p} + \mathrm{pn} + \mathrm{m}) \rightarrow \mathrm{O}(\mathrm{IT}(\mathrm{pm} + \mathrm{pn})) \tag{8}$$

Performance Appraisal of the IGWO-LP Algorithm on Benchmark Functions

23 benchmark functions (F1 to F23) are classic functions that have been used by many researchers (Mirjalili, Mirjalili, and Lewis 2014). The results of our algorithm were compared with the results of other metaheuristic algorithms such as GSA (Kennedy and Eberhart 1995), PSO (Rashedi, Nezamabadi-Pour, and Saryazdi 2009), GWO (Mirjalili, Mirjalili, and Lewis 2014), and FEP (Yao, Liu, and Lin 1999). Moreover, these benchmark functions are modified, rotated, expanded, and are a combination of classical functions that has the highest complexity amongst the current benchmark functions (Mirjalili, Mirjalili, and Lewis 2014). These benchmark functions are the minimum functions that can be divided into three groups: unimodal, multimodal, fixed-dimension multimodal, and composite functions.

The configured parameters of the IGWO-LP algorithm in our analysis for this section are as follows: The total number of wolves is 30 and the total number of repetitions in the algorithm is 200. The parameters z, k, and m are defined as 0.15, 2, and 4, respectively. The total number of repetitions and the parameters of other methods of GWO, PSO, GSA, and FEP are such as (Mirjalili, Mirjalili, and Lewis 2014). Our algorithm runs independently 30 times for each benchmark function. The average results shown in Table 3 suggest that the proposed algorithm is on average better than other comparable algorithms.

In F1 to F7, which are one-sided benchmark functions, IGWO-LP has a better response in all cases except F6. In the F8 function, GWO performs better than the rest of the algorithms, in F12 to F14, FEP is better in

Table 2. The NMI results for all datasets.

Dataset	Algorithms	NMI	
SYN-FIX	IGWO-LP	(Z = 3)	(Z = 5)
		0.971746	0.836166
	GLmetric	0.95258	0.79345
	DYNMOGA	0.91321	0.20945
	FactNet	0.87785	0.74654
SYN-VAR	IGWO-LP	0.95019	0.923129
	GLmetric	0.91342	0.85462
	DYNMOGA	0.89991	0.83277
	FactNet	0.39671	0.17731
Cell phone calls	IGWO-LP	0.965461	
	GLmetric	0.938625	
	DYNMOGA	0.480447	
Enron mail	IGWO-LP	0.946454	
	GLmetric	0.9104	
	DYNMOGA	0.6047	
Football	IGWO-LP	0.60143	
	GLmetric	0.59575	
	DYNMOGA	0.5739	

performance compared to other algorithms. In F15 to F19, our proposed algorithm performs better than the others. In the F20, the GSA performed better, in the F21 and F22, the GWO algorithm performed better, and in the F23 function, the GSA performed better.

Data Clustering Using Proposed Algorithm

We use 9 benchmark datasets from the UCI databases (Blake and Merz 1998), which are popular in cluster analysis literature. The chosen datasets have variety in terms of number of data objects, number of categories and number of attributes. Three quarter of data objects in each dataset is chosen randomly for training and the remainder is used for testing. Properties of these datasets are provided in Table 4.

Total number of search agents is 30 and total number of iterations is set at 500. The stopping criteria is reaching the total number of iterations. Results are compared to GSA (Bahrololoum et al. 2012), *k*-means (Nanda and Panda 2014), PSO (De Falco, Della Cioppa, and Tarantino 2007), K-PSO (Kao, Zahara, and Kao 2008), K-NM-PSO (Kao, Zahara, and Kao 2008), and Firefly Algorithm (Senthilnath, Omkar, and Mani 2011).

Results obtained from these methods are compared based on sum of the intra-cluster distances as well as Misclassification Rate. Misclassification rate is the percentage of misclassified instances on the total test set. To calculate this rate, first, the number of misclassifications is calculated. Since the actual class labels are known in the test set, it is possible to do so. Then the number of

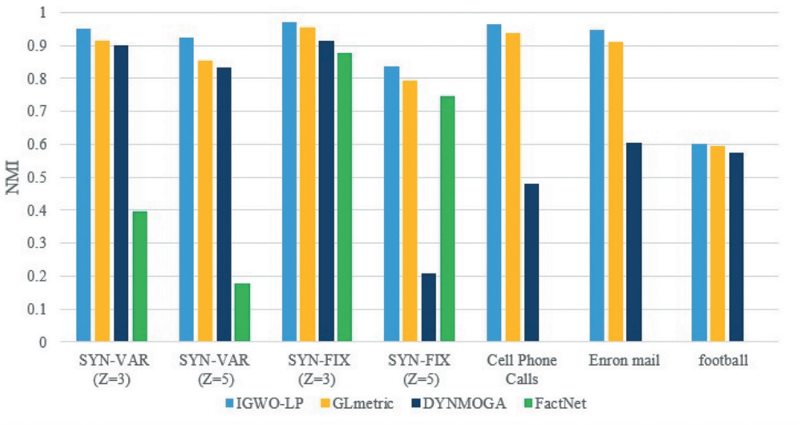

Figure 12. Comparing the results of algorithms using NMI criterion.

misclassifications is divided by total number of instances in the test set. Intra-cluster distances obtained from the 7 algorithms are provided in Table 5. The values are the average of the sums of intra-cluster distances over 25 runs.

In dataset Credit, Cancer-Int and Diabetes which have the highest number of data objects, IGWO-LP outperforms all algorithms while PSO, KPSO and K-NM-PSO have the worst results. Dermatology dataset has the highest number of classes which is 6 classes. Again, our proposed algorithm gives substantially better result on this dataset while PSO and FA have the worst results. The only datasets where IGWO-LP does not outperform other

Table 3. Results of benchmark functions (F1 to F23).

Function	*IGWO-LP*	*GWO*	PSO	GSA	FEP
F1	**5.7849e-36**	6.59e-28	0.000136	2.53e-16	0.00057
F2	**9.4232e-22**	7.18e-17	0.042144	0.055655	0.0081
F3	**6.4878e-07**	3.29e-06	70.12562	896.5347	0.016
F4	**1.2982e-09**	5.61e-07	1.086481	7.35487	0.3
F5	**4.8072**	26.81258	96.71832	67.54309	5.06
F6	0.66066	0.816579	0.000102	**2.5e-16**	0
F7	**0.0016112**	0.002213	0.122854	0.089441	0.1415
F8	−5516.8023	**−6123.1**	−4841.29	−2821.07	−1254.5
F9	**2.4632e-14**	0.310521	46.70423	25.96841	0.046
F10	**2.2797e-14**	1.06e-13	0.276015	0.062087	0.018
F11	**0.0029237**	0.004485	0.009214	27.70154	0.016
F12	0.03737	0.053438	0.006917	1.799617	**9.2E-06**
F13	0.45877	0.654464	0.006675	8.899084	**0.00016**
F14	4.556	4.042493	3.627168	5.859838	**1.22**
F15	**0.0002796**	0.000337	0.00577	0.003673	0.005
F16	**−1.0298**	−1.03163	−1.03163	−1.03163	−1.03
F17	**0.3975**	0.397889	0.397887	0.397887	0.398
F18	**2.6543e-05**	3.000028	3	3	3.02
F19	**−3.8604**	−3.86263	−3.86278	−3.86278	−3.8614
F20	−3.2659	−3.28654	−3.26634	**−3.31778**	−3.27
F21	−9.8072	**−10.1514**	−6.8651	−5.95512	−5.52
F22	−10.3963	**−10.4015**	−8.45653	−9.68447	−5.53
F23	−10.2731	−10.5343	−9.95291	**−10.5364**	−6.57

Table 4. Properties of benchmark datasets.

Dataset	Data objects	Classes	Attributes
Heart	303	2	35
E. Coli	327	5	7
Horse	364	2	58
Dermatology	366	6	34
Cancer	569	2	30
Balance	625	3	4
Credit	690	2	15
Cancer-Int	699	2	9
Diabetes	768	2	8

Table 5. Intra-cluster distances of each algorithm on UCI datasets.

Dataset	IGWO-LP	GSA	K-means	PSO	K-PSO	K-NM-PSO	FA
Heart	**1683.51**	4350.15	10284.16	2298.83	2144.42	6693.52	9095.10
E. Coli	**28.49**	571.97	675.21	40.37	408.93	566.36	57.78
Horse	15.58	14.85	19.69	**8.42**	15.10	70.41	94.37
Dermatology	**115.83**	654.98	442.87	1862.78	742.20	555.49	1448.48
Cancer	**19.53**	34.89	90.25	169.88	29.62	195.20	126.67
Balance	12893.22	131293.30	20137.76	61987.01	**10966.71**	17153.45	37640.44
Credit	**1369.16**	1398.57	3349.21	5090.77	1973.18	4194.19	4036.46
Cancer-Int	**96.72**	260.61	148.82	230.30	272.94	150.33	117.52
Diabetes	**134.60**	6747.03	156.52	3140.59	5148.62	9525.25	5488.42

Table 6. Average error rate of the algorithms in percentage.

Dataset	IGWO-LP	GSA	K-means	PSO	K-PSO	K-NM-PSO	FA
Heart	9.18	25.01	37.81	16.82	16.26	38.20	19.46
E. Coli	3.92	7.15	15.51	38.82	21.71	26.31	8.71
Horse	14.22	7.22	12.31	24.56	12.91	4.06	16.90
Dermatology	3.39	34.29	40.47	14.39	27.51	5.13	25.37
Cancer	5.04	12.72	14.62	13.74	6.37	11.77	29.44
Balance	7.27	3.38	15.11	11.34	36.87	24.19	2.59
Credit	10.70	14.66	27.67	18.05	36.16	33.33	24.09
Cancer-Int	14.76	10.09	26.06	8.41	5.76	25.86	7.92
Diabetes	2.40	7.59	38.30	12.47	2.62	30.76	31.54
Average	7.87	13.56	25.31	17.62	18.46	22.17	18.44
Rank	1	2	7	3	5	6	4

algorithms are Horse and Balance. For Horse dataset, PSO gives the best answer followed by GSA, K-PSO and our proposed algorithm, For Balance dataset, IGWO-LP has the second best answer after K-PSO.

Table 6 shows the mean error rate of each algorithm on each dataset over 25 simulation runs. For all datasets except Heart, Dermatology, Cancer-Int data sets, GSA provides an importantly smaller mean error rate compared to K-means and PSO. Although IGWO-LP in some datasets does not provide the lowest error rate, the average error rate of IGWO-LP over all datasets is 7.87% which is the lowest and puts it in rank number 1 followed by GSA and PSO. K-means and K-NM-PSO have significantly higher error rate compared to other algorithms.

Conclusion

In this article, a new multi-objective optimization method was introduced that combines the grey wolf optimization algorithm and the label propagation algorithm to identify communities in dynamic networks. In this way, we improved the grey wolf optimization algorithm using two approaches. The first approach is to improve the way that wolf move in the search space using local search by changing the way the population is updated. The second approach is that after a certain number of repetitions of the algorithm, a number of wolves are accidentally applied to the algorithm so that if the algorithm goes out of its way, it can be directed to the right and optimal path. We also gave the answers in each repetition of the proposed algorithm to the improved label propagation algorithm to find a better answer if possible.

The experiments with both artificial and real datasets have shown that the IGWO-LP algorithm performs better than other comparable algorithms in terms of quality and speed of detection. We also examined the proposed algorithm on 23 standard test functions (F1 to F23), which are known as benchmark functions, and as seen in the article, the proposed algorithm has better answers than the PSO, GSA, GWO, and FEP algorithms in half of the functions. At the end, the efficacy of the proposed algorithm was proved by comparing its performance with six other clustering methods on nine datasets from UCI machine learning laboratory. The mean error of IGWO-LP is 7.87% in the dataset that is the minimum between other algorithms. The bests next are GSA and PSO, respectively. According to the comprehensive evaluations and the mean clustering error rate, the proposed algorithm works better compared to the standard algorithms.

Notes

1. Normalized Mutual Information.
2. University of California, Irvine.
3. Improved Grey Wolf Optimizer Label Propagation.
4. Label Propagation.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

References

Bahrololoum, A., H. Nezamabadi-Pour, H. Bahrololoum, and M. Saeed. 2012. A prototype classifier based on gravitational search algorithm. *Applied Soft Computing* 12 (2):819–25. doi:10.1016/j.asoc.2011.10.008.

Besharatnia, F., A. Talebpour, and S. Aliakbary. 2020. Metaheuristic multi-objective method to detect communities on dynamic social networks. *International Journal of Computational Intelligence Systems* 1875–6883. 1875–6891; e. doi:10.2991/ijcis.d.210415.001.

Bilal S, and Moussaoui A, 2017, Evolutionary algorithm and modularity for detecting communities in networks, *Physica A: Statistical Mechanics and its Applications*, 473. DOI:10.1016/j.physa.2017.01.018

Blake, C. L., and C. J. Merz (1998). UCI repository of machine learning databases. Irvine: Department of Information and Computer Sciences, University of California. http://www.ics.uci.edu/mlearn/MLRepository.

Chi, Y., X. Song, D. Zhou, K. Hino, and B. L. Tseng. 2009. On evolutionary spectral clustering. *ACM Transactions on Knowledge Discovery from Data(TKDD)* 3 (4):17.

Danon, L., A. Dazguilera, J. Duch, and A. Arenas. 2005. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* 09:09008. doi:10.1088/17425468/2005/09/P09008.

Danon, L., Dazguilera A., Duch J., and Arenas A. 2005. Comparing community structure identification. *Journal of Statistical Mechanics Theory & Experiment* (9), 9008. http://dx.doi.org/10.1088/17425468/2005/09/P09008.

De Falco, I., A. Della Cioppa, and E. Taranti. 2007. Facing classification problems with particle swarm optimization. *Applied Soft Computing* 7 (3):652–58. doi:10.1016/j.asoc.2005.09.004.

Folino, F., and C. A. Pizzuti. 2010. multi-objective and evolutionary clustering method for dynamic networks. In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM*), Denmark, 256–63.

Folino, F., and C. Pizzuti. 2014. An evolutionary multi-objective approach for community discovery in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering* 99 (8):1–1. doi:10.1109/TKDE.2013.131.

Girvan, M., and M. E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99 (12):7821–26. doi:10.1073/pnas.122653799.

Kao, Y. T., E. Zahara, and I. W. Kao. 2008. A hybridized approach to data clustering. *Expert Systems with Applications* 34 (3):1754–62. doi:10.1016/j.eswa.2007.01.028.

Katoh, N., and T. Ibaraki. 1998. Resource allocation problems. *Handbook of Combinatorial Optimization* 2:159–260.

Kennedy, J., and R. Eberhart. 1995. Particle swarm optimization in neural networks proceedings. *IEEE International Conference On*, *4*(4), 1942–48.

Lin, Y. R., Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. 2008. Facetnet: A framework for analyzing communities and their evolutions in dynamic networks. In *The 17th International World Wide Web Conference(WWW08*), Beijing, 685–94. doi:10.1145/1367497.1367590

Lin, Y. R., Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. 2009. Analyzing communities and their evolutions in dynamic social networks. *Acm Transactions on Knowledge Discovery from Data* 307–08. doi:10.1145/1514888.1514891.

Lin, Z., X. Zheng, N. Xin, and D. Chen. 2014. CK-LPA: Efficient community detection algorithm based on label propagation with community kernel. *Physica A: Statistical Mechanics and Its Applications* 416(C), 386-399.

Liu., F., J. Wu, S. Xue, C. Zhou, J. Yang, and Q. Sheng. 2020. Detecting the evolving community structure in dynamic social networks. *World Wide Web* 715–33. doi:10.1007/s11280-019-00710-z.

Mirjalili, S., S. M. Mirjalili, and A. Lewis. 2014. Grey wolf optimizer. *Advances in Engineering Software* 69:46–61. doi:10.1016/j.advengsoft.2013.12.007.

Moayedikia, A. 2018. Multi-objective community detection algorithm with node importance analysis in attributed networks. *Applied Soft Computing*. doi:10.1016/j.asoc.2018.03.014.

Nam P. Nguyen, Thang N. Dinh, Yilin Shen, and My T. Thai, 2014, Dynamic Social Community Detection and Its Applications, *PLoS ONE* 9(4):e91431. DOI:10.1371/journal.pone.0091431

Nanda, S. J., and G. Panda. 2014. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation* 16:1–18.

Pizzuti, C. 2012. A multiobjective genetic algorithm to find communities in complex networks. *IEEE Transactions on Evolutionary Computation* 16 (3):418–30. doi:10.1109/TEVC.2011.2161090.

Pourkazemi, M., and M. R. Keyvanpour. 2017. Community detection in social network by using a multi objective evolutionary algorithm. *Intelligent Data Analysis*. doi:10.3233/IDA-150429.

Rashedi, E., H. Nezamabadi-Pour, and S. Saryazdi. 2009. GSA: A gravitational search algorithm. *Information Sciences* 179:2232–48. doi:10.1016/j.ins.2009.03.004.

Ronhovde, R. K., R. Peter, and Z. Nussinov. 2013. An edge density definition of overlapping and weighted graph communities. *arXiv Preprint arXiv* 1301:3120.

Samie, M. E., and A. Hamzeh. 2016. Community detection in dynamic social networks: A local evolutionary approach. *Journal of Information Science*. doi:10.1177/0165551516657717.

Senthilnath, J., S. N. Omkar, and V. Mani. 2011. Clustering using firefly algorithm: Performance study. *Swarm and Evolutionary Computation* 1 (3):164–71. doi:10.1016/j.swevo.2011.06.003.

Xinzheng, N., S. Weiyu, and C. Q. Wu. 2017. A label-based evolutionary computing approach to dynamic community detection. *Computer Communications*. doi:10.1016/j.com.2017.04.009.

Yao, X., Y. Liu, and G. Lin. 1999. Evolutionary programming made faster. *Evolutionary Computation IEEE Transactions On* 3:82–102. doi:10.1109/4235.771163.

Zare, A., F. Hamidi, and A. Rahati. 2015. Improved grey wolf algorithm. In *4th Joint Congress of Iranian Fuzzy and Intelligent Systems*.

Zhu, X., and Z. Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Accessed July 20, 2020. http://www.cs.cmu.edu/~enron.