

## Research Article

# Metaheuristic Multi-Objective Method to Detect Communities on Dynamic Social Networks

Fatemeh Besharatnia, AliReza Talebpour<sup>\*</sup>, Sadegh Aliakbary<sup>ID</sup>

Faculty of Computers Science and Engineering, Shahid Beheshti University, Tehran, 1983969411, Iran

## ARTICLE INFO

### Article History

Received 18 Jun 2020  
Accepted 28 Jan 2021

### Keywords

Dynamic networks  
Community detection  
Heuristic algorithms  
Grey Wolf optimize

## ABSTRACT

Community detection is an important area in social networks analysis, which has many applications. Most social networks are inherently dynamic, consisting of constantly changing communities and therefore, community detection is a challenge in such networks. Since the communities in dynamic networks change, we need high-performance methods for community detection which observe the network changes and update the detected communities, instead of finding the communities for each network snapshot from scratch. As a result, the need to incremental community detection algorithms has been emerged. In this paper, a novel method is presented to identify communities in dynamic social networks, based on a multi-objective metaheuristic algorithm using label propagation technique, in order to detect communities incrementally. The evaluation of the proposed method, which includes examination of different artificial and real networks, shows that the proposed algorithm outperforms the state-of-the-art algorithms with respect to modularity and normalized mutual information (NMI) objectives.

© 2021 The Authors. Published by Atlantis Press B.V.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

## 1. INTRODUCTION

In the current decade, online social networks are developing with an increasing growth. Some of these networks now have hundreds of millions of members, and such networks play an inevitable role in human life today. One of the most important characteristics of social networks is the existence of communities, and community detection is still an interesting research area in the domain of social networks analysis. With the help of community extraction methods, common group behaviors can be categorized and the each part of the network can be analyzed and studied.

A social network is often expressed in a graph in which the vertices are actors and edges indicate their relationship. Analyzing social networks is a significant research area which has attracted the researchers' attention in recent decades. Particularly, the study of dynamic social networks which their structure changes over the time. Identifying communities in such dynamic networks is a challenging issue. In order to recognize the communities in dynamic networks, algorithms are mainly divided into independent community mining and incremental community mining [1]. In the first group, communities are independent in each snapshot and extracted regardless of alterations over the time and their association with the communities in the past. This approach is appropriate for unstable networks. In the second group, the problem of identifying the communities in each snapshot is solved by using the

previous communities' information related to the previous snapshots. This approach benefits better efficiency and time complexity because it utilizes information used to identify the communities of the previous snapshots and therefore, it is more appropriate for the networks that gradually vary [1,2]. In this paper, we consider the second approach of community detection in dynamic networks.

In this paper, the identification of dynamic communities is presented as a multi-objective optimization problem which integrates the improved Grey Wolf Optimizer (GWO) algorithm with the improved label propagation (LP) algorithm and the incremental community mining approach. The GWO algorithm is a metaheuristic method that we use at community detection in networks. Besides, we improve the algorithm at two points making the community detection to function better. To detect dynamic communities, two objectives must be considered. The first is the quality of the structure of communities at the  $t$  moment. We considered the Modularity [3] criteria to evaluate this criterion. The second is the temporal cost (TC) to evaluate the similarity between communities of successive snapshots, for which the normalized mutual information (NMI) criterion developed by Danon *et al.* is used [4]. In other words, the first objective is to maximize modularity which shows that how sufficiently the clustering discovered shows the data at the present time, and the second objective is to optimize NMI metric which admeasures the distance among two clustering at successive intervals.

The experiments were performed on both artificial and real data sets and the results indicated that our proposed algorithm

<sup>\*</sup>Corresponding author. Email: [talebpour@sbu.ac.ir](mailto:talebpour@sbu.ac.ir)

outperforms the baseline algorithms. Some advantages of the proposed algorithm are as following:

1. Our proposed algorithm integrates improved multi-objective GWO algorithm and improved LP algorithm creating better performance and more accuracy than the compared algorithms.
2. The suggested solution is related to a linear time complexity by considering the number of edges.

The structure of the rest of this paper is as follows: The second section covers a review of the related works. In the third section, the suggested method is presented. The fourth section shows the evaluation results and a comparison with state-of-the-art algorithms and, finally, the fifth section covers the conclusion and the future works.

## 2. LITERATURE REVIEW

The detection of communities is considered as one of the basic methods in obtaining the hidden structure in the network. A lot of algorithms have been presented in last two decades to detect community structure which is a demanding task, including statistical inference [4–6], dynamic LP [7–9], spectral clustering [10], information-theoretic methods [11,12], topology based methods [13,14] and modularity optimization [15–17].

Reguvan *et al.* introduced LP algorithm which has a linear time complexity, it is also one of the fastest algorithms for large networks [8].

The main problem mostly observed in community detection methods is ignoring the time complexity of the algorithm. This is while social networks are usually dynamic and they change over the time. In recent years, researchers investigated community detection through dynamic social networks. There are two different approaches to solve this problem: incremental community mining and independent community mining. The independent community mining procedure is often used to evaluate the graph and the communities during the time. Various methods are suggested for this approach including heuristic algorithms and community mining. Wang *et al.* proposed a kernel-based algorithm to follow the evolution of community in dynamic networks, which depends on the main (kernel) nodes to propagate evolutionary relations among communities in different snapshots. This method has no parameters and is appropriate for discovering the points of division and integration [18]. Takaffoli *et al.* proposed a structure in order to design and detect the community evolution. They identified communities in each snapshot with a “community flag,” a new modification, and their proposed two-step MODEC algorithm [19]. In Sun *et al.* [20], an alteration of modularity density (D value) was used to analyze the community cohesion that considers both the links among vertices and link weights.

The second approach, incremental community mining, utilizes the information taken from the earlier snapshots to extract communities in each Snapshot. Niua *et al.* suggested a label-based multi-objective genetic algorithm (L-DMGA) to identify dynamic communities. The algorithm applied a genetic algorithm for the purpose of improving two goals (the quality of the clustering and the TC). Furthermore, a LP method is applied to reset the

network and limit mutation conditions, which makes it more efficient and effective [21]. Han *et al.* presented an algorithm called adaptive label propagation algorithm (ALPA), which computes the whole graph for community detection, and is applied only to those parts of the graph including active nodes unlike the other methods, which results in the computational cost reduction. Another advantage of this algorithm is that there is no need to initial parameter [22]. Folino *et al.* introduced a new multi-objective genetic algorithm called DYNMOGA in the problem of community detection in dynamic networks, and they used the Pareto optimality theory to find the answers since their algorithm was multi-objective. DYNMOGA employed nondimensional sorting to make a balance between the quality of TC and instantaneous momentum. Owing to the lack of scalability, the algorithm has a long load and run in large networks [23].

Lin *et al.* presented a FacetNet framework as one of the most popular algorithms and used a multi-objective scenario to detect the communities in dynamic social networks. Further, it used the cost function proposed by Chakraborti *et al.* FacetNet is one of the algorithms which organizes the evolution of the community structure [14]. Samie *et al.* introduced a two-step method which in order to detect community in social networks, uses global and local information, that firstly examines the global information, and a new algorithm is employed to identify communities with high accuracy in the second step. This algorithm uses the local information and mines deeply [2]. The structural clustering algorithm for networks (SCAN) was enhanced by Aston *et al.*, which is considered to be one of the algorithms that detects communities in static networks to update a local structure in less time. In their new algorithm called DSCAN (dynamic SCAN), there was no need to set parameters like SCAN [24].

Dakiche *et al.* predicted the evolutionary community in dynamic social networks by the characteristic changes of a community during the evolution of its life cycle. Their tests were conducted on the DBLP and Facebook datasets, and regarding the characteristics of the influential members and structural features of the community, they predicted the next event which might happen for a community with a very high accuracy [25]. Zhao *et al.*, in their paper, suggested an incremental method for community detection using handling subgraphs. Firstly, a comprehensive analysis was done and then four types of incremental elements along with various updating strategies were proposed. Finally, in order to detect community in dynamic networks, the algorithm was presented [26]. Zhang *et al.* suggested a new dynamic method for identifying communities in their study. According to authors, this method has different advantages such as considering a community as a mixture of topics, generating latent variables using Gibbs sampling and detecting community and its topic as well as its temporal changes [27]. Liu *et al.* in their paper proposed a new migration operator to enhance detection rate in evolving community structures using classic genetic operators. This operator is applied on a new hybrid algorithm.

A new direct modularity calculation method was also presented which expands the search space [28]. Li *et al.* presented a new method to identify communities by which the community ownership of partial vertices was evaluated gradually based on a vertex-based metric titled permanence, which is from incremental identification, to prevent all the nodes in the network to be

reassigned to their related communities [29]. Pérez-Peló *et al.*, in order to detect communities in social networks, presented an algorithm based on Greedy Randomized Adaptive Search Procedure (GRASP) methodology which is also metaheuristic type. The authors modeled the problem of community detection as an optimization problem, so that the goal is to optimize the modularity of the network [30]. Cheraghchi *et al.*, proposed a novel granular approach to detect dynamic communities in social networks. They suggested the micro- and macrogranules in two levels of nodes and communities in which microgranules are considered as input and the output would be meaningful communities. The suggested algorithm detects the structure of communities and exploits the number of communities according to the interactions in the network [31].

Modularity and homogeneity are two factors which were optimized simultaneously by a multi-objective optimization evolutionary mechanism suggested by [32]. Modularity is a criterion of the community structure that measures the thickly connections inside the communities. Optimizing modularity leads to structure clustering realization. Homogeneity is a novel criterion that categorizes the attributes with the aim of nodes having homogeneous attribute value inside every community and having unlike attribute values in various communities.

Community score and community fitness were maximized in order to find the Pareto front using a multi-objective evolutionary algorithm suggested by [33] for community detection. The authors obtained a set of optimal solutions using multi-objective genetic algorithm and the best one was selected utilizing modularity function. To certify the effective connections of the nodes throughout the genetic operators in the network, they utilized the locus based adjacency representation. Moreover, to make sure of the population's diverseness, the uniform crossover was used.

<sup>1</sup> normalized variation of information

In Zhuang and Chang [34], authors suggested a new modularity-based dynamic community detection algorithm, called DynaMo, which can effectively identify communities of dynamic networks even better than using static algorithms. This adaptive and incremental method maximizes the modularity gain and updates the community structure simultaneously. In Choi and Choi [35], the method of the algorithm to improve the community structure is discovering the most important community nodes in the network over time.

Authors of [36] proposed a new clustering community detection algorithm based on the PSO-LDA model. Gibbs sampling method was used to map the quantitative parameters from semantic information to semantic space, since the semantic model is LDA model. After that, the overlapping community detection was solved using a PSO strategy. At the end, the semantic communities which were detected were analyzed by semantic modularity (SimQ).

DECS is the name of a new multi-objective evolutionary clustering algorithm suggested by authors which uses a migration operator along with other effective operators to group the nodes and their neighbors. Then, in order to extend the search space, the structure information of networks was encoded using a genome matrix. Another aim of this algorithm is optimizing the modularity that is done based on the genome matrix [37].

In Jokar and Mosleh [38], authors suggested a new algorithm based on the LPA and the intrinsic structure of the network which is link density feature. According to the results of comparison between the suggested algorithm and the basic LPA, the output results of the proposed method are more certain and stable. Moreover, although the proposed method has better time complexity compared to most of the methods, it is similar to the basic LPA. A summary of the algorithms described in this section is provided in Table 1.

**Table 1** Comparing different algorithms for detecting community in dynamic social networks.

| Algorithm Full Name Title of the Paper  | Community Detection Methods                                | Objective Function   | Approach    | The Number of Algorithms Compared                       | The Number of Datasets Evaluated |
|---|--|--|-------------|---|----------------------------------|
| Metaheuristic multi-objective method to detect communities on dynamic social networks | Meta-heuristic (Grey Wolf Optimizer and label propagation) | NMI, error rate and modularity   | Incremental | L-DMGA, DYN-MOGA, FacetNet and ALPA                     | 5                                |
| Community evolution of social network: feature, algorithm and model [18].             | Core-based Algorithm of Tracking Community Evolution.      | The correlation coefficients between community size and age (GROWTH) as well as between evolution trace span and member stability (METABOLISM) | Independent | _____   | 7                                |
| A framework for analyzing dynamic social networks [19].                               | Event-based methods  | Form, dissolve survive, split, merge and mutual topics   | Independent | Event-based frameworks including Asur, Palla and Greene | 2                                |
| Maximizing modularity intensity for community partition and evolution [20].           | Modularity-based methods                                   | Modularity intensity   | Independent | D-value method  | 2                                |
| A label-based evolutionary computing approach to dynamic community detection [21].    | Meta-heuristic (genetic algorithm and label propagation)   | NMI and error Rate   | Incremental | DYNMOGA and FacetNet                                    | 6                                |

*Continued*

**Table 1** | (Continued)

|  |  |   |             |  |   |
|--|--|---|-------------|--|---|
| Community detection in dynamic networks via adaptive label propagation [22]                    | Adaptive label propagation algorithm   | NMI, NVI <sup>1</sup> and Jaccard index, modularity and time  | Incremental | FacetNet, ILCD and Infomap                                 | 3 |
| An evolutionary multi-objective approach for community discovery in dynamic networks [23]      | Meta-heuristic (genetic algorithm)   | NMI and error rate  | Incremental | FacetNet,  | 6 |
| Facetnet: A framework for analyzing communities and their evolutions in dynamic networks [14]. | Probabilistic methods  | Mutual information and modularity   | Incremental | EvolSpec, FacetNet, NCut and SNMF                          | 4 |
| Community detection in dynamic social networks: A local evolutionary approach [2].             | Meta-heuristic (genetic algorithm)   | NMI   | Incremental | DYNMOGA, FacetNet and Kim-Han                              | 6 |
| An incremental method to detect communities in dynamic evolving social networks [26].          | Incremental method   | Modularity  | Incremental | LabelRankT, FacetNet, IncOrder and ARTISON                 | 3 |
| Multi-objective community detection method by integrating users' behavior attributes [32].     | Structure clustering technology and attribute categorization technology simultaneously | Modularity, the maximum, average and standard deviation of best modularity values (Qmax, Qavg, Qstd) and best homogeneity values (Hmax, Havg, Hstd) | Independent | _____  | 6 |
| DynaMo: Dynamic community detection by incrementally maximizing modularity [34].               | Modularity-based methods   | Modularity  | Incremental | Louvain, Batch, DynaMo, QCA, Gre-Mod, LBTR-LR and LBTR-SVM | 7 |

### 3. PRELIMINARIES

In this section, we will first define communities in dynamic social networks and also Pareto optimum in multi-objective optimization, then the problem statement and finally the fitness function would be explained.

#### 3.1. Dynamic Communities

Consider a dynamic social network model of the series of  $G_1; G_2; \dots; G_n$ , where  $G_i$  represents the network at the  $i^{\text{th}}$  snapshot. While every snapshot can be formed like a graph, that includes a collection of nodes ( $V_i$ ) and edges ( $E_i$ ), which is demonstrated by  $G_i = (V_i, E_i)$ . In addition, consider  $C_i = \{C_i^1, C_i^2, \dots, C_i^{n_i}\}$  which  $C_i$  communities discovered at the  $i^{\text{th}}$  snapshot where community  $C_i^p \in C_i$  is a graph shown by  $(V_i^p, E_i^p)$  [2].

#### 3.2. Pareto Optimality

To recognize communities in a static network, the problem can be posed like a single-objective problem, but it is a multi-objective problem in dynamic networks, since in this type of networks, two qualities of the community should be investigated (snapshot quality [SC] and TC). Also, as already mentioned, our problem is a multi-objective optimization one. Although no optimal answer can be found in this type of problem, a set of answers which are appropriate for the decision maker is always available. A solution for solving these subjects is to transform the problem into an objective using the equation [39].

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

where (SC) shows the snapshot quality and (TC) is considered as the temporal cost.  $SC(G_i; C_i)$  evaluates the detected communities quality,  $C_i$  is according to the present graph,  $G_i$ , and the similarity between the present communities is detected by  $TC(C_{i-1}; C_i)$ ,  $C_{i-1}$  are to the communities which are identified in the prior timestamp,  $C_{i-1}$ . The  $\alpha$  ( $0 \leq \alpha \leq 1$ ) parameter is applied for balancing both of the cost functions. Employing Eq. (1), an optimal community can be discovered by the algorithm. A major constraint of this method is related to the amount of  $\alpha$ , which in order to make a balance between two functions, it has to be used in the role of an input parameter. Therefore, obtaining the optimal value for the  $\alpha$  parameter is a challenging issue in many cases. For this purpose, we used Pareto optimal theory. Regarding the static  $G_i$  network, a multi-objective problem  $(\Omega, F_1, F_2, \dots, F_n)$  can be explained as [2]:

$$\min F_i(C^{ni}), i = 1, \dots, n \text{ subject to } C^{ni} \in \Omega \quad (2)$$

where  $\Omega = \{C_i^1, C_i^2, \dots, C_i^{n_i}\}$  is a possible cluster set of  $G_i$  in the  $i^{\text{th}}$  snapshot and  $F = \{F_1, F_2, \dots, F_n\}$  is a set of criteria functions of the  $n$  unit. Each  $F_i : \Omega \rightarrow R$  is a target function which specifies a possible clustering.  $F$  is considered as a collection of competitive goals and should be optimized concurrently. Although any specific answer could not be found for this problem, a series of answers could be obtained by Pareto optimal theory [40]. Considering the two solution  $C_1$  and  $C_2 \in \Omega$ ,  $C_1$  is dominant in  $C_2$  if only:

$$\forall i : F_i(C_1) \leq F_i(C_2) \wedge \exists \text{is.t. } F_i(C_1) < F_i(C_2) \quad (3)$$

Now, we have to recognize that which objective function has to be deteriorated in order to improve another one, that a nondominated solution is applied for this task. This nondominated solutions

is named “Pareto optimal.” In other words, for a multi-objective optimization, there is a collection of Pareto optimal, nondominated solutions that leastwise one of them is better compared to others in each set. This method aims at discovering the Pareto optimal set. F vector transforms the space of solution to the space of target. “Pareto front” is the name of the dominant solutions that are designed in the space of target. So, the targets can be solved by the Pareto Front completely. In fact, the solutions in Pareto optimal set cannot be compared to each other and none of them is the best. So it can be said that Pareto front solutions try to satisfy all the objectives. Furthermore, an input parameter is not essential in the latter method. Therefore, it was decided to be utilized in the proposed algorithm.

### 3.3. Problem Statement

For community detection in dynamic social networks, we combine two algorithms, one of which is GWO that is a metaheuristic algorithm which we improved it. The next is LP algorithm that we use per repetition of GWO algorithm. So we named our proposed algorithm as MOGWO-LP (multi-objective GWO LP) that is multi-objective. This algorithm aims optimizing both SC and TC without requiring to balance factor  $\alpha$  in advance. These two objectives can be properly satisfied by the solutions consisting the Pareto front of the multi-objective optimization. Also we assume that the communities do not overlap and all the nodes are of the same value.

### 3.4. Fitness Functions

According to Eq. (1), the SC and TC are considered by the algorithm for the COST optimization. The modularity criteria are used for SC, which is a very popular function. In addition, it is generally utilized to assess the quality of the communities which are identified by the detection algorithms. It is used to indicate clustering quality in each snapshot as follows [41]:

$$Q = \sum_{s=1}^k \left[ \frac{l_s}{m} - \left( \frac{d_s}{2m} \right)^2 \right] \quad (4)$$

In each snapshot, the graph includes  $k$  community as  $C = \{C_1, C_2, \dots, C_k\}$ . Here,  $m$  indicates the number of edges of the entire network in every snapshot, and  $l_s$  shows the number of edges associated with the nodes of the  $C_s$  community, and  $d_s$  is considered as the summation of the degree of nodes. If the  $Q$  values are close to 1, the partitions have solid community structure and on the other hand, if the values of  $Q$  are near to 0, the partition is not according to the structure community.

The NMI criterion was used for TC which indicates the variance betwixt the present snapshot and the prior one [4]. In fact, using NMI, the similarity between the produced partition by the algorithm and the ideal and considered partition can be detected which is a criterion for community performance evaluation. According to the previous studies and assuming the existence of the basic information regarding the community, this measure has proper reliability and is commonly utilized for evaluating these kind of algorithms. Assume that there are two parts:  $A = \{A_1, \dots, A_a\}$  and  $B = \{B_1, \dots, B_b\}$  related to a network in communities. Consider  $C$

as the confusion matrix which its element  $C_{ij}$  shows the number of nodes that are both in the communities of  $A_i \in A$ , which are available in the community  $B_j \in B$ . The NMI ( $A, B$ ) can be calculated by:

$$NMI(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log(C_{ij}N/C_i C_j)}{\sum_{i=1}^{C_A} C_i \log\left(\frac{C_i}{N}\right) + \sum_{j=1}^{C_B} C_j \log\left(\frac{C_j}{N}\right)} \quad (5)$$

In Eq. (5),  $N$  represents the number of nodes and  $C_A$  ( $C_B$ ) shows the number of communities in the partition  $A$  ( $B$ ),  $C_i$  ( $C_j$ ) illustrates the summation of the elements of  $C$  in row  $i$  (column  $j$ ), NMI is determined in the interval  $[0, 1]$  and is maximal as the communities which are compared are the same. Also, If  $A = B$ ,  $NMI(A, B) = 1$ .  $NMI(A, B) = 0$  if  $A$  and  $B$  are entirely varied.

In addition, the error rate criterion was used to analysis the algorithms. Since we have information provided and collected for the community memberships at every snapshot, the error rate of the community structures attained through various algorithms has been studied straightly. For the purpose of calculating the error rate [23], an  $n \times k$  indicator matrix  $Z$  was created, so that  $n$  represents the number of nodes and  $k$  is considered as the number of communities. Furthermore, an analogous indicator matrix  $G$  is applied to demonstrate the real community. Error rate can be calculated by:

$$\|ZZ^T - GG^T\| \quad (6)$$

Eq. (6), computes the spacing between community  $Z$  and real community  $G$ .

## 4. THE PROPOSED METHOD

### 4.1. MOGWO-LP Algorithm

The optimizer grey wolf algorithm is inspired from nature, which is proven to be more accurate than similar methods like genetic algorithms and particle swarm optimization (PSO) algorithm [42]. In this paper, this algorithm is used to identify community in dynamic social networks, which is improved in two ways, leading to better performance. The pseudocode of this improved algorithm is illustrated in Figure 1, where the (a) parameter is obtained from Eq. (2), and the equations of the  $A, C$  parameters are also available in [42].

**First Approach:** In optimization algorithms, there should be a balance between exploring the search space and convergence toward the overall answer. In the GWO algorithm, the values of  $\vec{a}$  and  $\vec{A}$  parameters control these two phases. In the classical GWO algorithm, (a) parameter decreases linearly from 2 to 0:

$$\vec{a}(t) = 2 - \frac{2t}{\text{Max\_iter}} \quad (7)$$

In this equation,  $t$  is the number of current iteration and  $\text{Max\_iter}$  is the maximum number of iterations of the algorithm. For example, if the maximum number of iterations is 100, in the first iteration ( $t = 0$ ), the value of  $\vec{a}$  will be equal to 2. In subsequent iterations, the value of  $\vec{a}$  decreases linearly and finally in the last iteration ( $t = 99$ ), its value reaches 0.

**Input:** A dynamic network  $G = \{G_1, G_2, \dots, G_T\}$ , and the number  $T$  of snapshot

**Output:** A clustering  $C_t = \{C_{t1}, C_{t2}, \dots, C_{tk}\}$  for each network snapshot  $G_t$

**Phase 1:** Algorithm1( MOGWO-LP ) [42]

```

1: Set the grey wolf population  $X_i$  ( $i = 1, 2, \dots, n$ )
2: Set  $a$ ,  $A$  and  $C$ 
3: Describe the  $MN$  = maximum number of iterations
4: Evaluate the fitness value of each search agent // search agent = The Grey wolf
5:  $X_\alpha$  = best search agent
6:  $X_\beta$  = second best search agent
7:  $X_\delta$  = third best search agent
8: while  $t < MN$  do
9:   for each search agent do
10:    Update the place of the existing search agent by
11:   end for
12:   Update  $a$ ,  $A$  and  $C$ 
13:   Assess the fitness of all search agents
14:    $t = t + 1$ 
15: end while
16: return  $X_\alpha$  as best solution

```

**Phase 2:**



```

1: for  $t=2 : T$  do
2:   if  $S \bmod t == \emptyset$  then
3:     while there is a node in the main active node list do
4:       if  $y == p$  then //  $p$  is the second component of the ordered pair in the
           main active node list
5:         Do not accept the nodes with this condition
6:       end if
7:     end while
8:     Identification of active nodes and placing them in the main active node list and the active node list related to
           the same snapshot.
9:     Applying the MOGWO-LP algorithm on active nodes
10:    if  $R$  was a multiple of  $t$  then
11:      Applying the MOGWO-LP algorithm on the entire graph
12:    end if
13:    Returning the solution  $C_t = \{C_{t1}, C_{t2}, \dots, C_{tk}\}$  of the Pareto Front with the maximum
           amount of modularity value
14:  end for

```

**Figure 1** | Model and pseudocode of MOGWO-LP algorithm.

Since the search process of the GWO is nonlinear and very complex, the linear reduction of this parameter cannot show that how the real search process is done. Therefore, to improve this problem, (a) parameter is made non-linear, and a new formula is proposed as follows:

$$\vec{a}(t) = 2 - 2 \left( \frac{t}{\text{Max\_iter}} \right)^k \tag{8}$$

The nonlinearity of this formula causes a balance between these two phases, and the algorithm presents better answers, where, k is a positive integer. For values between 0 and 1, the algorithm will focus more on local search, but general search may not be sufficient. For the value of 1, the equation becomes the linear equation of the classical algorithm, and for more than 1, there will be more focus on the general search, and after a proper and thorough exploration of the search space, the convergence operation will be performed. Figure 2 shows the graph of  $\vec{a}$  for the values of k equal to 0.5, 2 and 1.

**Second Approach:** Therefore, to better explore the space around the best answer obtained, a mapping is suggested to better converge based on the location of the best wolf ( $X_a$ ). This mapping converts the location of the best answer (wolf) to a new area, and if a better answer is found in the new area, that answer is selected as the best answer of the algorithm. This new area can be defined by:

$$\vec{X}_n = \vec{X}_\alpha + r (U_b - L_b) (Z - 0.5) \tag{9}$$

This local search of  $\vec{X}_n$  is performed with the  $\vec{X}_\alpha$  center and r radius.  $U_b$  and  $L_b$  are the upper and lower range of search space. In addition, z is the mapping parameter calculated by the following function:

$$z_{t+1} = m \times z_t \times (1 - z_t) \tag{10}$$

where m variable controls the mapping. Moreover, to improve the answer, after every repetition of the MOGWO algorithm, the obtained answer sets are improved by Label propagation which was improved using Jaccard Index in reference [43]. Accordingly, it is said that LP algorithm [8] does not provide answers with high accuracy, since it first randomly labels the nodes. Jaccard Index is used to have a better performance. First, the weight of each edge is determined based on the resemblance among the communications between the two vertexes. The similarity between the two vertexes is calculated according to the similarity criterion presented below.

$$W_{i,j} = \begin{cases} J(V_i, V_j) + 1 & e(V_i, V_j) \in E \\ 0 & e(V_i, V_j) \notin E \end{cases} \tag{11}$$

similarity value between the vertexes by the Jaccard index, and in the of a relationship between the  $V_j$  and  $V_i$  vertexes, 1 is also added to calculate their direct and indirect similarity. For the  $V_j$  and  $V_i$  vertexes, the Jaccard index is computed by:

$$J(V_i, V_j) = \frac{|V_i \cap V_j|}{|V_i \cup V_j|} \tag{12}$$

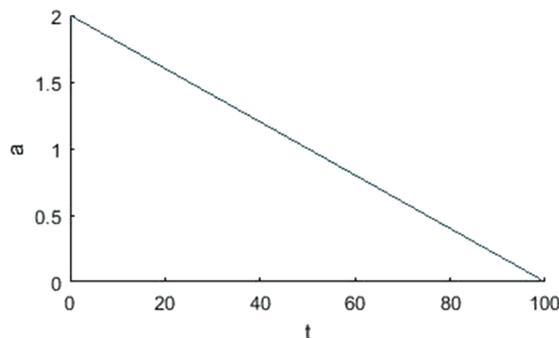


Figure 2a | Reduction functions of parameter of  $\vec{a}$  for different values of k (k = 1).

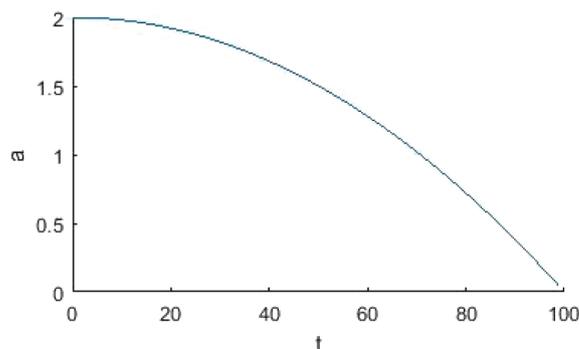


Figure 2b | Reduction functions of parameter of  $\vec{a}$  for different values of k (k = 2).

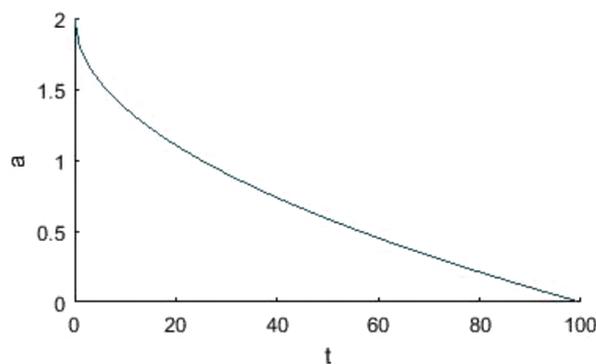


Figure 2c | Reduction functions of parameter of  $\vec{a}$  for different values of k (k = 0.5).

where the number of common vertexes is in numerator and in denominator, there is the total number of vertexes associated with  $V_j$  and  $V_i$  vertexes. When the similarity values among all vertexes attached to the graph is calculated, the graph can be transformed into a weighted graph. In the next stage, the LP algorithm is exerted to this graph. In this case, the algorithm uses the vertex with highest weight to select the next vertex among the neighboring vertexes, and this approach provides more stability and more satisfactory results. In addition, the higher the similarity between two vertices according to Formula 11, the more likely two vertices are to be in the same community, and vice versa. The figure of our proposed algorithm can be seen in Figure 1 (phase1).

## 4.2. MOGWO-LP Algorithm for Community Detection

Our suggested approach, as said before, is an incremental one which has two phases where the first phase is performed only on the first snapshot. In this phase, MOGWO-LP Algorithm operates on the total network and detects the communities. In the second phase, that is from the second snapshot on, the suggested algorithm for community detection is performed only on those parts of the graph which have active nodes. Of course for improving the answers in very few next snapshots, we operate the algorithm on the total network. Finally in our proposed algorithm, a collection of solutions are produced which all of them are Pareto front and every of them is related to a balance among two objective functions, also every partition in the network has various number of communities. As mentioned before, Pareto Fronts can satisfy both objectives simultaneously. The solution with the highest modularity value is selected as the result of GT that is the best approach for measuring the community structure as yet. The suggested model design and semi-code algorithm may be observed in Figure 1.

### 4.2.1. Portion 1

The label of an active node is not the main label between its neighbors and it would change its label in the case of updating. The idea of detecting the active nodes is based on [22]. Finding the active nodes has six modes: First Mode: Add an edge. A) The added edges should be within the community. Do not activate the nodes because they are inside the community. B) The added edge should be outside the community. Activate both communities' nodes that are associated with the new edge. Second Mode: Remove an edge. A) The removed edge should be outside the community. Nodes are not activated. Removing the edge makes the structure of communities more obvious. B) The removed edge should be within the community because the removing may transform the community into smaller sections and these sections can connect to other communities. Therefore, the target nodes are added to the list of active nodes. Third mode: Add a single node. Make a new community for it. Fourth mode: Connect a node with multiple edges. Divide the procedure to two stages, first add a single node, and then add the edges attached to one by one (first mode). The target nodes are put in the active node list. Fifth mode: Remove a single node. Then, the structure of the recent group remains unchanged. Sixth mode: Remove a node with multiple connected edges. First, all the connected edges are removed one by one (second mode). Then the node itself is removed. (This removal may turn the community into small sections, and these sections can connect to other communities.)

### 4.2.2. Portion 2

Moreover, we propose two approaches for MOGWO-LP Algorithm to better detect the communities in the network. According to the results of experiments, these approaches perform better in detecting communities. Approach one: an active node list is set for each snapshot and the active nodes of the same snapshot are put within it. Further, all active nodes in all of the snapshots are saved in another active node list (main active node list). There is an ordered pair  $(x, y)$  in each of its nodes. The number of the node is represented by

$x$  and  $y$  is the score of each node. Furthermore, at first  $y = 0$ . When an active node is found in each snapshot each time, if that node causes the community to change, which means to divide the community into smaller communities or integrate with a larger community, its  $y$  will increase one unit, and if it does not change,  $y$  will decrease one unit. In addition, it will be back to the main active node list (Figures 3 and 4).

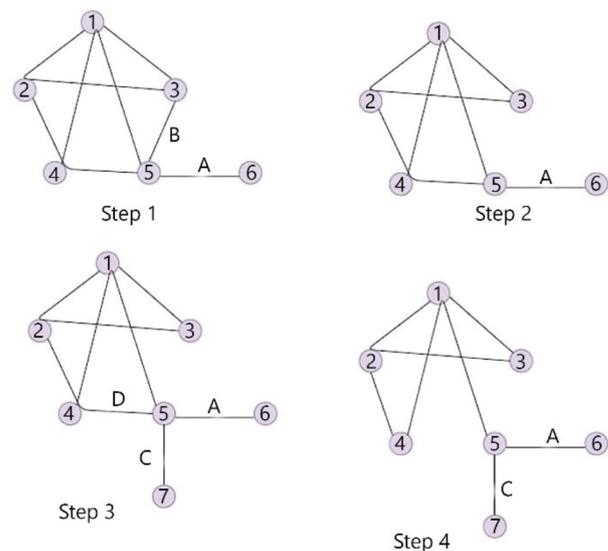
After  $S$ , the snapshot of the active nodes is checked in the main data structure. The reason why the ineffective active nodes (for example, for dataset 1,  $y = -2$  or  $-7$ ) are not considered in the algorithm even if they become active again in next snapshots, is that they probably will not affect the structure of communities such as not dividing community into smaller communities or merging communities.

Approach two: in order to improve some snapshots' operations, we perform the proposed algorithm on the entire graph (the condition that is line 9 of the second phase in the algorithm). We suggest this approach due to the fact that if we want to operate the algorithm on the active nodes in all the snapshots and save the first one, the algorithm would underperform and the experimental results show that this is a good idea.

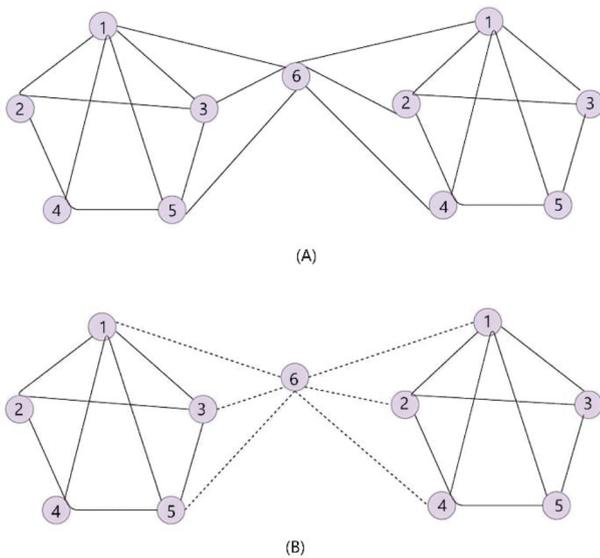
Finally, the algorithm presents a solution collection that are all Pareto Front. Pareto Front solutions are nondominated ones which their main features are optimal SC and TC. The solution with the greatest modularity value is picked out to be the best solution, that is the finest way for measuring community structure.

## 5. RESULTS AND DISCUSSION

In the following part, the performance of the MOGWO-LP algorithm is evaluated by DYNMOGA, FacetNet, ALPA, L-DMGA algorithms, and the results are compared with different types of artificial and real data.



**Figure 3** | Active node (node 5) that is ineffective: (Step 1: Add node 6 and edge A, Step 2: Delete edge B, Step 3: Add node 7 and edge C, Step 4: Delete edge D).



**Figure 4** | Active node (node 6) that has an effect: (A: Add node 6 with its edges, B: Remove node 6 with its edges).

The MOGWO-LP algorithm is developed in MATLAB. The parameters pertaining to the algorithms are considered as follows: the number of repetitions is 100, the population and the number of wolves is 100, the average result over 50 runs measure, the parameter  $z$  equals to 0.15,  $m$  is 4, and  $k$  is 2, the mutation rate is 0.1 and crossover rate is 0.9. The elite reproduction is set to have 50% of the size of population. For DYNMOGA, the parameters are set according to [22], where the crossover rate is 0.8 and mutation rate is 0.2. Other parameters are the same as the parameters of L-DMGA algorithm. The ALPA algorithm does not require an initial parameter. In addition, two verification functions including NMI and error rate were exerted for the algorithms evaluation. NMI and error rate are described in Section 3.4.

## 5.1. Results on Synthesized Networks

### 5.1.1. Dataset 1

Two kinds of datasets, SYN-VAR and SYN-FIX, proposed by Girvan and Newman [3] are applied and then used in [44] and [45]. 128 nodes in SYN-FIX network contains four communities, that every one of them gets 32 nodes. The average degree of every node is 16. In addition, every node possesses  $z$  (3 and 5) edges connected to the nodes which are outside the community. In this dataset, there are 10 snapshots. Further, in each snapshot, and in each community, 3 nodes accidentally exit the community and attach to the other communities.

The SYN-VAR dataset is consisted of 256 nodes, that are classified to 4 communities, each composed of 64 nodes. This dataset, similar to SYN-FIX, contains 10 snapshots. For the second snapshot, 8 nodes are selected from every community to create a new community with 32 nodes, which this action is repeated until the fifth snapshot. Then, in the next 5 snapshots, the nodes come back into the primary communities. So, the number of communities for 10 snapshots is equal to 4, 5, 6, 7, 8, 8, 7, 6, 5, 4, respectively. Each node has the average degree of the half size of that community, and 16 nodes are omitted by accident per snapshot and 16 new nodes are added.

Figure 5 shows the results of NMI criterion for the SYN-FIX and SYN-VAR data sets when  $z = 3$  and  $z = 5$  and  $S = 3, 6$  and  $R = 4, 8$  and  $y = -2$  or  $-7$ . In most snapshots, the MOGWO-LP algorithm has a higher NMI value than the other algorithms, and it is one in some snapshots, which indicates that it has a more accurate structure than the other algorithms.

Figure 6 displays the error rate for these two datasets with the mentioned characteristics. The proposed algorithm has a lower mean error rate compared to other algorithms. Besides, the entire average NMI results and the error rate of the Algorithms for dataset SYN-FIX, and SYN-VAR are illustrated in Tables 2 and 3.

### 5.1.2. Dataset 2

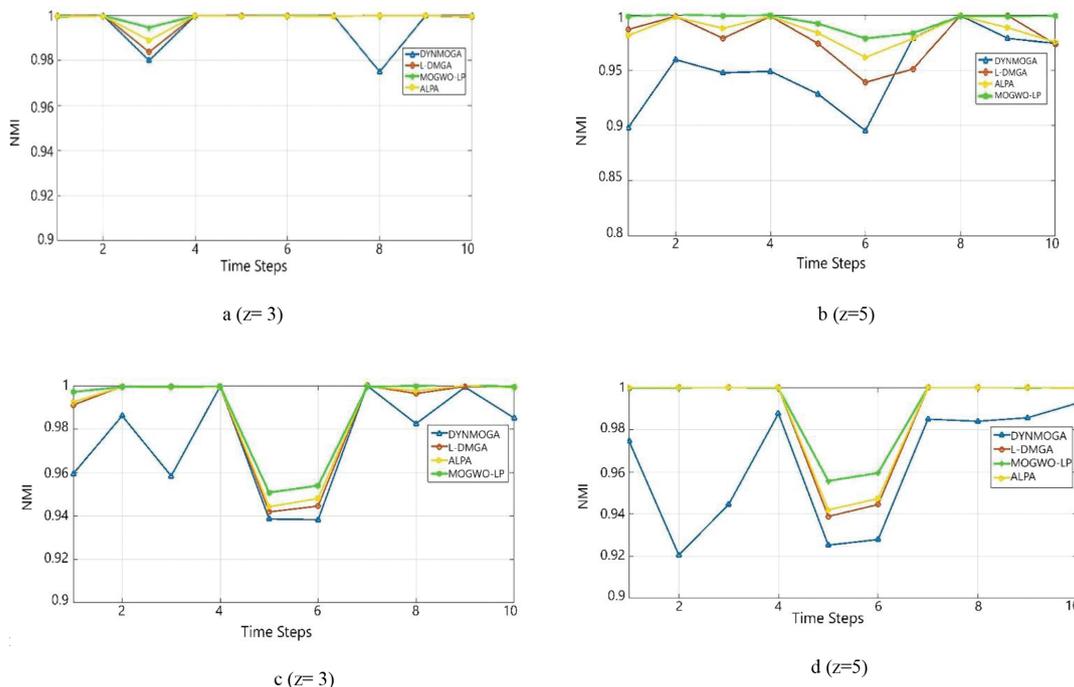
In the presented study, the benchmark adopted by Lin *et al.* [14] was used for generating the first dataset, which has similarity to Girvan and Newman dataset in [3].

Every graph includes 128 nodes which are classified to 4 communities that every one of them contains 32 nodes. A fixed AvgDegree is involved by each node along with  $z$  edges that are connected to the outside nodes of the community. The community structure becomes fuzzy through increasing  $z$  value. In this regard, AvgDegree value and the  $z$  value are two distinct values. First, it signifies a sparse and dense network when  $\text{avgDegree} = 16$  and 20, respectively. Second, it makes a graph to be of a clear and blurred community structure when  $z = 5$  and 6, respectively. The edges which were generated are more likely connecting to the nodes which are in the same community than nodes that are from different communities, in fact, the related probability of such edges is dependent on the value of  $z$ . In addition, we moved  $nC\%$  nodes out of one community to another in order to emulate dynamic networks in the experiment. Here, two scenarios are taken into consideration. First in this case, when 10% of nodes are selected by random of their prior communities and appointing to other communities, structure of the community remains relatively stable. On the other hand, there is a huge difference between the community structures and the original communities when 30% of nodes are chosen by random out of the prior communities which they had and appointing to other communities.

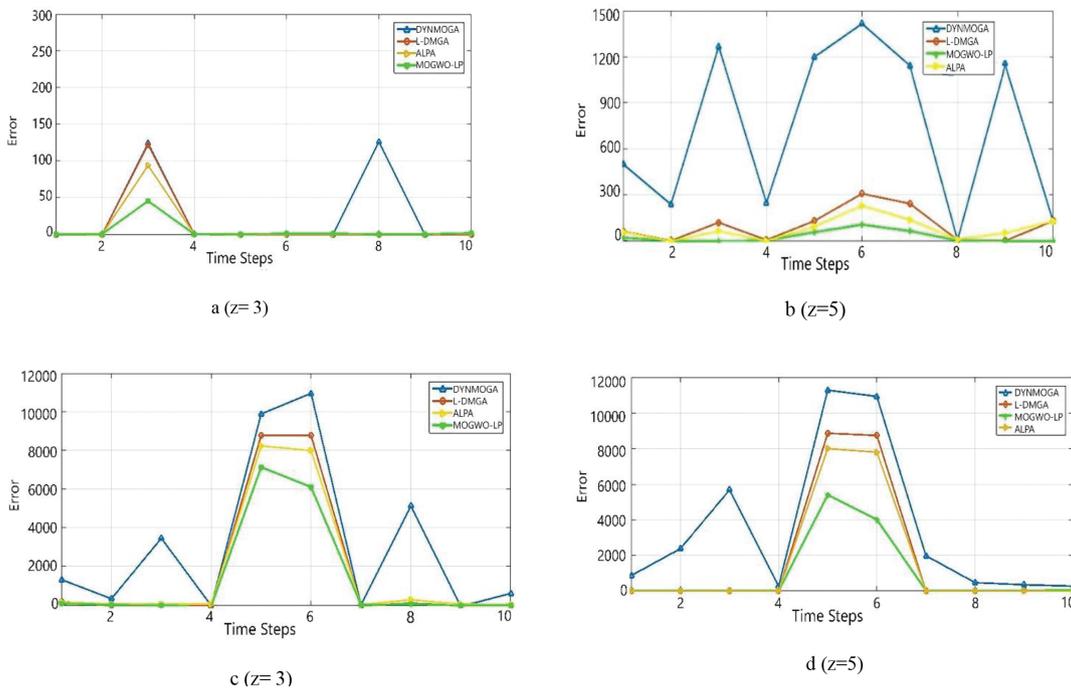
Figures 7 and 8 display the NMI and error rate. As shown in NMI, MOGWO-LP is higher than the rest of the algorithms in four types of networks. Moreover, MOGWO-LP has a lower error rate in most of the snapshots in comparison with the rest algorithms. Tables 4 and 5 demonstrate the average NMI and the error rate of dataset2 when  $\text{avgDegree}$  is 16.

Figures 9 and 10 demonstrate the NMI plot and error rate produced by MOGWO-LP and DYNMOGA and L-DMGA. Figure 9 illustrates the NMI, with  $\text{avgDegree}$  of 20 at the time (a)  $z$  is 5 with the percentage of node changes  $nC = 30\%$ , and when (b)  $z = 6$ , the percentage of node changes  $nC = 30\%$ . The error rate is plotted in Figure 9 using the same configuration.

Figure 9 shows NMI while the average degree is 20 and the percentage of the node replacements is  $nC = 30\%$  with two different assumptions: (a)  $z = 5$  and (b)  $z = 6$ . Likewise, the error rate using the pervious configuration is illustrated in Figure 10.



**Figure 5** | The NMI result of the SYN-FIX and SYN-VAR datasets. a and b SYN-FIX, c and d SYN-VAR.



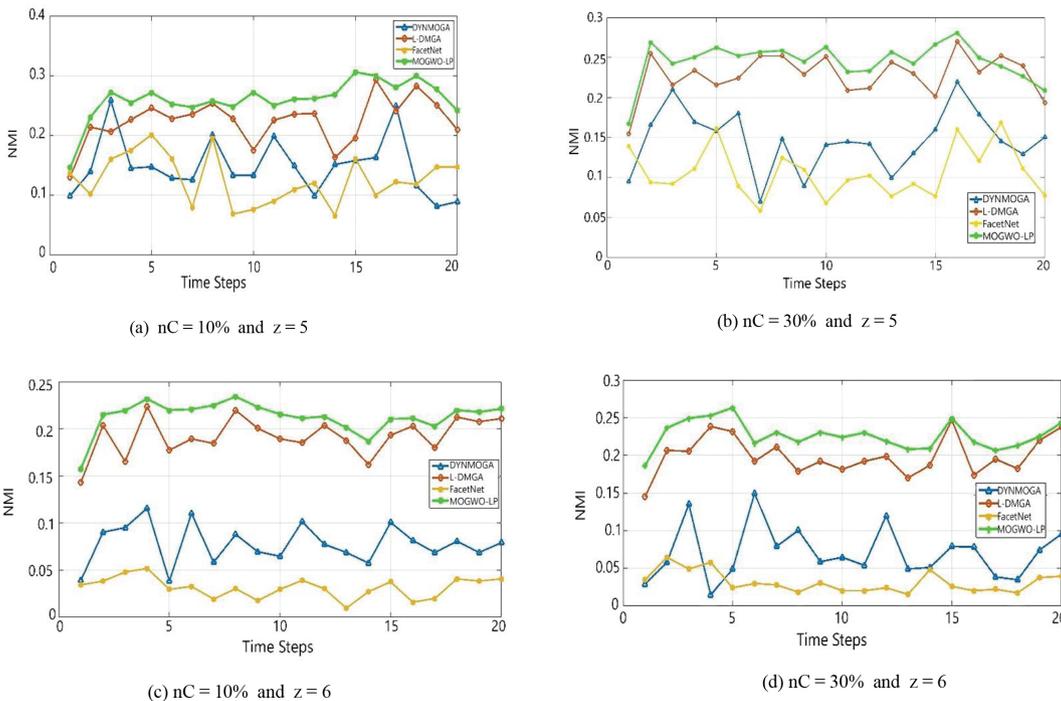
**Figure 6** | Error rate result of the SYN-FIX and SYN-VAR datasets. a and b SYN-FIX, c and d SYN-VAR.

**Table 2** | The average NMI results for dataset SYN-FIX and SYN-VAR.

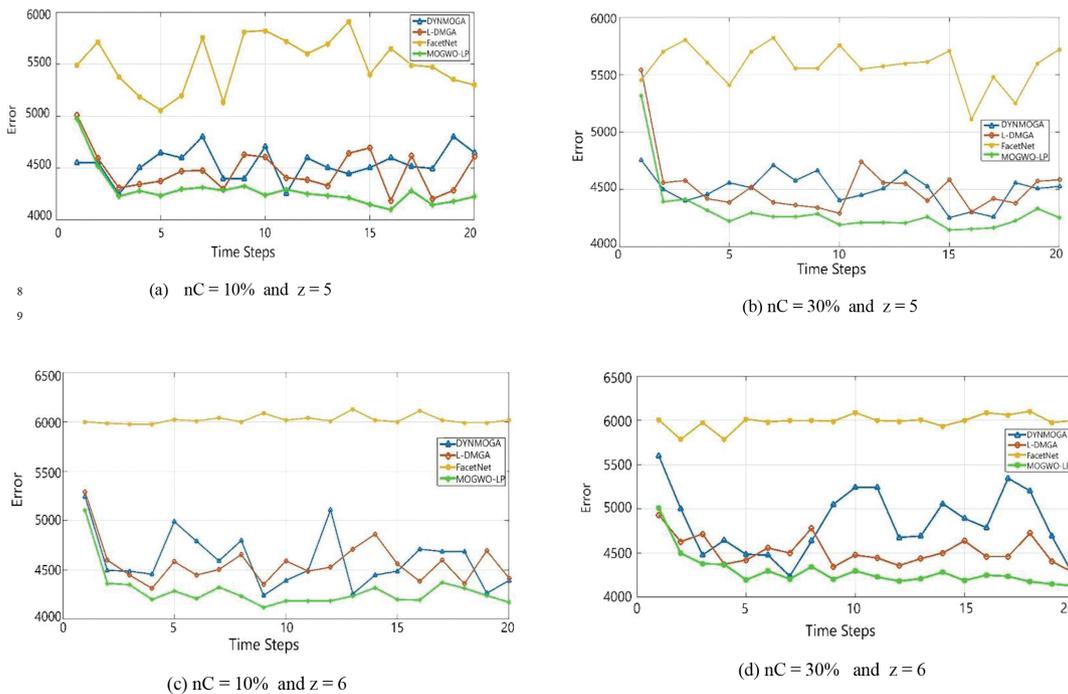
| Algorithms | SYN-FIX<br>(Z = 3) | SYN-FIX<br>(Z = 5) | SYN-VAR<br>(Z = 3) | SYN-VAR<br>(Z = 5) |
|------------|--------------------|--------------------|--------------------|--------------------|
| MOGWOLP    | 0.99929            | 0.99480            | 0.98999            | 0.99148            |
| L-DMGA     | 0.99827            | 0.97993            | 0.98719            | 0.98830            |
| ALPA       | 0.99872            | 0.98523            | 0.98797            | 0.98892            |
| DYNMOGA    | 0.9955             | 0.95092            | 0.97475            | 0.96282            |

**Table 3** | The average error rate results for dataset SYN-FIX and SYN-VAR.

| Algorithms | SYN-FIX<br>(Z = 3) | SYN-FIX<br>(Z = 5) | SYN-VAR<br>(Z = 3) | SYN-VAR<br>(Z = 5) |
|------------|--------------------|--------------------|--------------------|--------------------|
| MOGWOLP    | 5.05228            | 24.27263           | 1325.39411         | 959.14734          |
| L-DMGA     | 11.42384           | 99.50495           | 1760.2649          | 1768.21192         |
| ALPA       | 9.49062            | 75.84              | 1682.33239         | 1580               |
| DYNMOGA    | 24.53642           | 728.71287          | 3154.96688         | 3452.98013         |



**Figure 7** | NMI on dataset 2 when  $S = 4, 8, 12$  and  $R = 5, 10, 15$  and  $y = -2$  or  $-7$  or  $-15$  and  $avgDegree = 16$ .



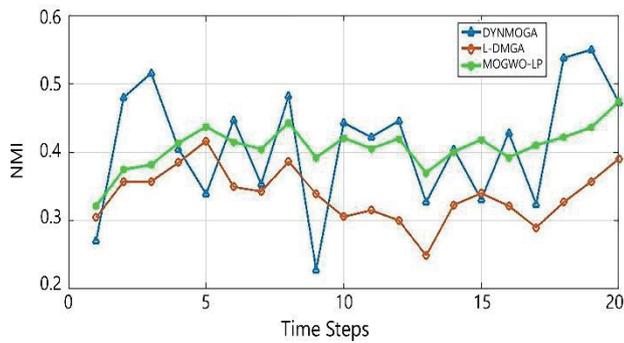
**Figure 8** | Error rates on dataset 2 when  $S = 4, 8, 12$  and  $R = 5, 10, 15$  and  $y = -2$  or  $-7$  or  $-15$  and  $avgDegree = 16$ .

**Table 4** | The average NMI of dataset2 when avgDegree is 16.

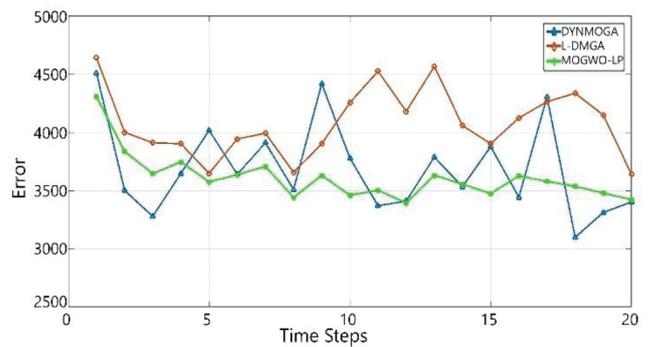
| Algorithms | Z = 5    | Z = 5    | Z = 6    | Z = 6    |
|------------|----------|----------|----------|----------|
|            | nC = 10% | nC = 30% | nC = 10% | nC = 30% |
| MOGWO-LP   | 0.25999  | 0.245291 | 0.21290  | 0.226358 |
| L-DMGA     | 0.22394  | 0.22835  | 0.192093 | 0.19919  |
| FaceNet    | 0.12726  | 0.10658  | 0.03155  | 0.03144  |
| DYNMOGA    | 0.14853  | 0.14640  | 0.077676 | 0.070611 |

**Table 5** | The average error rate of dataset2 when avgDegree is 16.

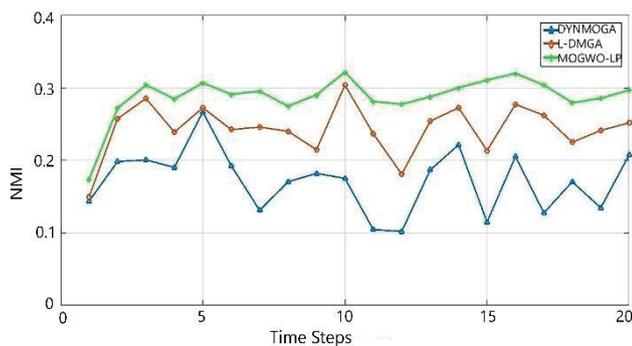
| Algorithms | Z = 5    | Z = 5    | Z = 6    | Z = 6    |
|------------|----------|----------|----------|----------|
|            | nC = 10% | nC = 30% | nC = 10% | nC = 30% |
| MOGWO-LP   | 4289.424 | 4306.522 | 4288.009 | 4292.978 |
| L-DMGA     | 4474.204 | 4524.444 | 4567.277 | 4522.293 |
| FaceNet    | 5506.051 | 5578.73  | 6019.904 | 5988.455 |
| DYNMOGA    | 4539.49  | 4505.079 | 4599.124 | 4834.793 |



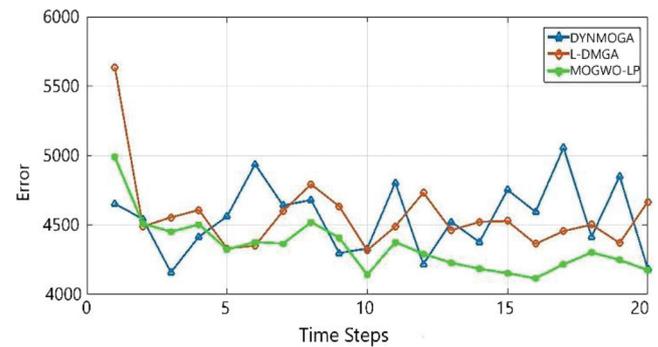
(a) nC = 30% and z = 5



(a) nC = 30% and z = 5



(b) nC = 30% and z = 6



(b) nC = 30% and z = 6

**Figure 9** | NMI on dataset 2 when S = 4, 8, 12 and R = 5, 10, 15 and y = -3, -8, -14 and avgDegree = 20.

**Figure 10** | Error rates on dataset 2 when S = 4, 8, 12 and R = 5, 10, 15 and y = -3 or -8 or -14 and avgDegree = 20.

The network becomes denser through  $avgDegree = 20$ . Regarding NMI, DYNMOGA is a bit finer compared to MOGWO-LP and L-DMGA whereas  $z = 5$ . While, MOGWO-LP can discover finer community structure when  $z = 6$  as it is presented by Figure 9(b). In Table 6, you can see the average NMI and the error rate in dataset2 when  $avgDegree$  is 20 and  $nC = 30\%$ .

### 5.1.3. Dataset 3

In order to develop the Girvan and Newman benchmark, Lancichinetti *et al.* [46] suggested a new benchmark, called Lancichinetti-Fortunato-Radicchi (LFR) model, by presenting power law degree distributions and various community size.

This LFR model includes 1000 nodes with  $avgDegree$  of 20. The highest node degree is 50, community size distribution is  $-1$ , the mixing parameter is 0.3. Moreover, the degree distribution has exponential number of  $-2$ . Approximately 70% of nodes in the suggested network have a degree under 20 and just one node has the highest degree. 10% of communities were selected accidentally for 5 time steps, and at the time steps of 2 and 4, they were split and merged at the time steps of 3 and 5, repeatedly. Figure 11 shows the results of the normalized cross-data and Error rate and also the mean NMI and Error rate for the LFR shown in Table 7, indicating that the MOGWO-LP algorithm performed better on all snapshots.

## 5.2. Results on Real-World Networks

### 5.2.1. Dataset 4

Cell phone calls dataset [47]: This dataset contains the record of phone calls between the fictitious Paraiso movement members in June 2006 within 10 days (there are 10 snapshot in this dataset). Each node stands for a specific mobile number and each edge presents a phone call between two mobile phones. The number of nodes is 400, and the exact information in terms of time and day of the phone call is recorded for every edge. The five important nodes in this network are as follows: Ferdinando Catalano (node 201) and his brother Estaban (node 6), David Vidro (node 2) and his two brothers Jorge and Juan (nodes 3 and 4). The stated popular nodes modified their phone call numbers during 7 and 8 days. Hence, during the days of 8–10, their node numbers changed from 201, 6, 2, 3, 4–301, 307, 310, 361, and 398, one after the other. In order to find more details about this dataset, see Folino and Pizzuti [23]. Since the real structure of the community is not recognized, the same technique utilized by Lin *et al.* in [24] was followed.

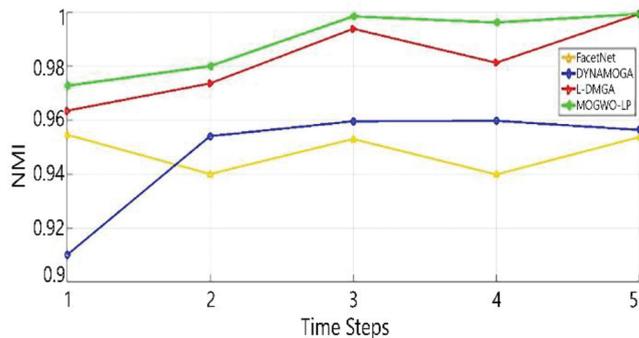
First, the MOGWO-LP algorithm was applied to the entire network, after which the community structure was obtained by the suggested algorithm as the real community structure. Then the error rate and

NMI were calculated by the compared algorithms on the network. As observed in Figure 12, MOGWO-LP has better NMI and a lower error rate in all snapshots, suggesting that it can create a more accurate community structure in dynamic networks.

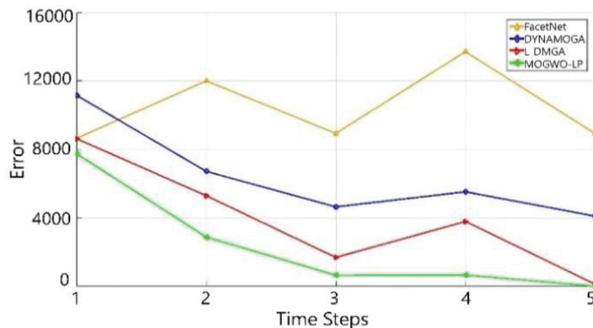
### 5.2.1. Dataset 5

Enron mail dataset [48]: This dataset includes email communications belonged to a US company in 1999–2002. The raw dataset is consisted of 517431 emails such as 151 users distributed and 3500 files which are reduced to 50,000 after the necessary modifications according to [23]. In categorizing the data in 2001, based on the months of the year, 12 sub-datasets were obtained and 6 communities were in network. A similar method was applied in the phone call dataset.

Figure 13 displays the results of NMI and the error rate for the Enron mail dataset when  $S = 3, 6, 9$  and  $R = 3, 7$  and  $\gamma = -2$  or  $-7$ . As shown in Figure 13(a), the MOGWO-LP algorithm except for



(a)



(b)

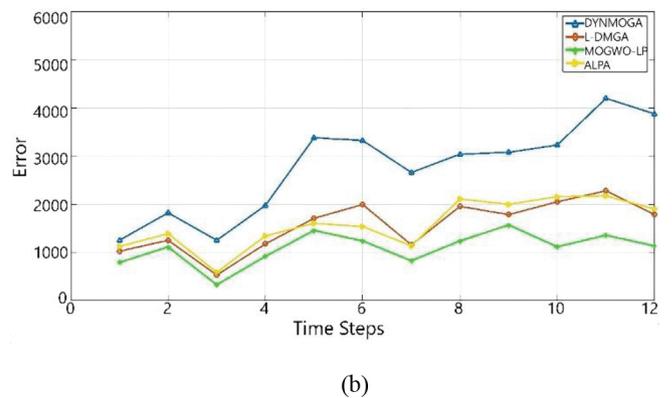
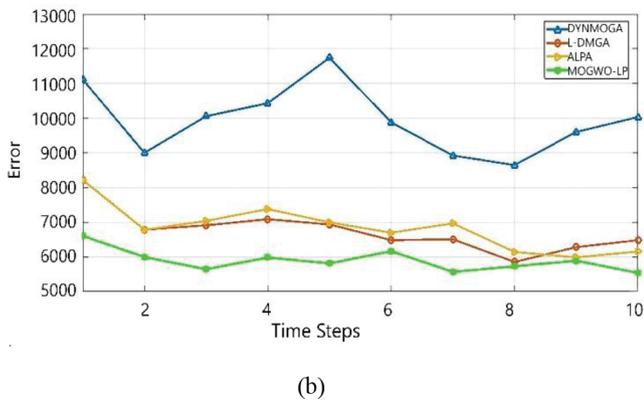
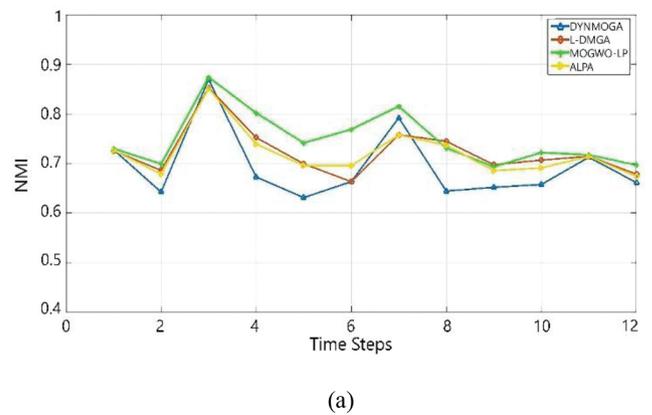
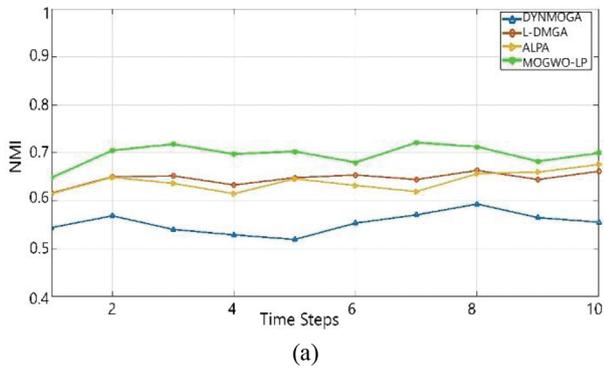
Figure 11 | The NMI result of LFR (a) and Error rate result of LFR (b), when  $S = 4$  and  $R = 3$  and  $\gamma = -2$ .

Table 6 | The average NMI and the error rate in dataset2 when  $avgDegree$  is 20 and  $nC = 30\%$ .

| Algorithms | Z = 5   |          | Z = 6   |          |
|------------|---------|----------|---------|----------|
|            | NMI     | Error    | NMI     | Error    |
| MOGWO-LP   | 0.40759 | 3608.81  | 0.28772 | 4361.333 |
| L-DMGA     | 0.33769 | 3686.885 | 0.24318 | 4569.967 |
| DYNMOGA    | 0.41023 | 4079.098 | 0.17124 | 4549.505 |

Table 7 | The average NMI and error rate in LFR.

| Algorithms | NMI      | Error    |
|------------|----------|----------|
| MOGWO-LP   | 0.989361 | 2392.243 |
| L-DMGA     | 0.982327 | 3907.638 |
| FaceNet    | 0.947975 | 10449.73 |
| DYNMOGA    | 0.948259 | 6431.261 |



**Figure 12** | The NMI result of Cell Phone Calls dataset (a) and Error rate result of Cell Phone Calls dataset (b), when  $S = 4, 8$  and  $R = 3, 7$  and  $\gamma = -2$  or  $-7$ .

**Figure 13** | (a) The NMI result of the Enron mail dataset and (b) Error rate result of the Enron mail dataset.

the 8 and 9 snapshots, has a better performance than the other algorithms. According to Figure 13(b), the proposed algorithm benefits from less error rate in all snapshots compared to other algorithms.

In Table 8, you can see the average NMI and error rate in Enron mail and cell phone calls dataset.

### 5.3. Discussion

In summary, in terms of NMI criterion, which is a maximizing problem, the obtained results show that for all of the aforementioned datasets in all of the mentioned states, the MOGWO-LP algorithm has a higher value compared to other algorithms. In addition, in terms of error rate criterion, which is a minimizing problem, the obtained results show that for all of the aforementioned datasets in all of the mentioned states, the MOGWO-LP algorithm has a lower value compared to other algorithms. These results show that the proposed algorithm is better in comparison to other algorithms since one of the best forecasters of further membership in real communities is previous community membership, using the NMI evaluation metric, we can say that MOGWO-LP is more efficient at identifying the real community structure.

We obtained the population and the number of wolves and the number of repetitions based on the comparable algorithms such as L-DMGA, and also the other parameters of the MOGWO-LP algorithm such as  $z, m, k$  were obtained by trial and error procedure and presents optimum for all case studies.

**Table 8** | The average NMI and error rate in Enron mail and cell phone calls dataset.

| Algorithms | Cell Phone Calls |          | Enron Mail |          |
|------------|------------------|----------|------------|----------|
|            | NMI              | Error    | NMI        | Error    |
| MOGWO-LP   | 0.69661          | 5895.095 | 0.74902    | 1092.212 |
| L-DMGA     | 0.64643          | 6755.975 | 0.72302    | 1558.544 |
| ALPA       | 0.64019          | 6837.3   | 0.72019    | 1588.826 |
| DYNMOGA    | 0.55369          | 9943.396 | 0.69369    | 2756.329 |

One of the strengths of our proposed algorithm is its scalability, which makes our approach to perform very well for very large networks, and the second advantage of our approach is that its complexity is linear that is a positive point; also, the third advantage is that initial parameters are not essential in the LP algorithm.

The fourth advantage is that the MOGWO-LP algorithm, by non-linearizing the parameter  $\vec{a}$ , creates a balance among search space exploration and convergence toward the general response. Search space exploration means finding new areas in the search space that population diversity is a momentous factor in it. In the convergence toward the answer, an accurate answer is expected. In the classical gray wolf optimization algorithm, these two phases are controlled by the value of the parameters  $\vec{a}$  and  $\vec{A}$ . The large amount of  $\vec{A}$  performs the general search and the small amount of  $\vec{a}$  performs the local search or convergence to the answer. Proper selection of the parameter  $\vec{a}$  can create a balance between general and local search, and this causes a proper and complete exploration of the search

space, which increases the NMI criterion and decreases the error ratio criterion.

Last advantage: One of our ideas was to check the MOGWO-LP algorithm on the whole graph in some snapshots for detecting communities (the situation which is line 9 of the second phase in the algorithm). As can be seen in the results, in most of the graphs in the snapshots where the proposed algorithm was applied to the whole graph, the algorithm performed better and in some cases even the MOGWO-LP algorithm performed better in relation to the prior snapshots and after the considered snapshot. This shows that if we use the idea of active nodes, in some snapshots, the MOGWO-LP algorithm still needs to be done on the whole graph to identify communities, so that the proposed algorithm works better. This idea, especially when the dataset is large, such as the Enron mail dataset, leads to performance improvement.

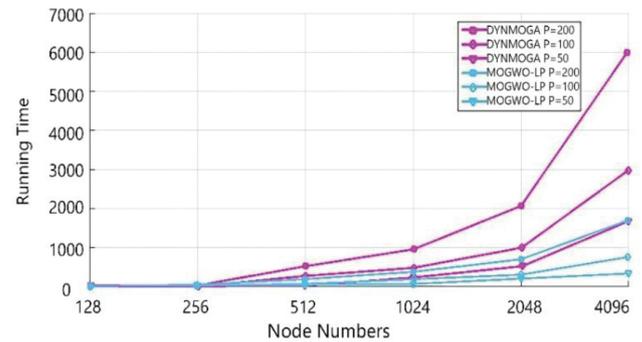
One of the disadvantages of our algorithm is trapping in the local optimal solution as other meta-heuristics algorithms. In this respect, however, we should note that the time of trapping in local optimal in our algorithm is less than the other meta-heuristics algorithms. In other algorithms, all members of the population converge to that member in the population that have the best answer. For example, in the PSO algorithm, all particles move to the particle that has the best position, and if the particle with the best position is near the local optimum solution, it causes the rest of the particles to converge to the same local answer. Another disadvantages of MOGWO-LP algorithm is when the community is unstable and dense, for example, for dataset 2 when ( $nC = 30\%$  and  $avgDegree = 20$  and  $z = 5$ ), our proposed algorithm does not work well since our algorithm is incremental. This method is more suitable for stable datasets. Of course, it should be noted that the other algorithms also did not performed well for this dataset.

However, in our proposed algorithm, in the initial iterations of the algorithm, some members of the population are exploring the search space so that even with the best answers near to local optimal, there is a chance that other members move to a better answer by exploring the nonindependent search space from the best of the current answers. In successive iterations, the algorithm changes from the problem space search phase to the convergence phase for the best solution to explore the final iterations around the best answer. Using the proposed label-correction algorithm along with the GWO Algorithm makes the search space searching even better by producing answers that will not be generated using the GWO.

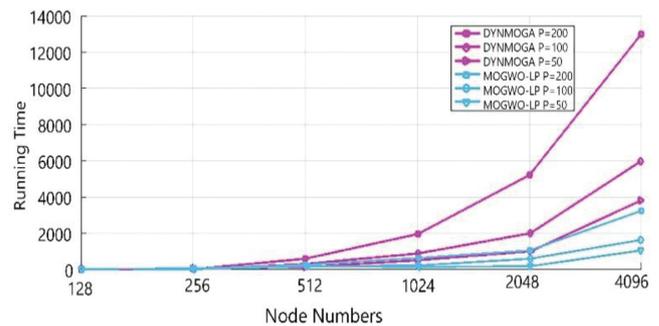
The performance of all algorithms for the first dataset is good compared to the rest of the datasets, and the performance of all algorithms for the second dataset has not been good compared to the rest of the datasets, even the MOGWO-LP algorithm. Also, in real datasets, although the Enron mail dataset has 50,000 nodes and the Cell Phone Calls dataset has 400 nodes, all algorithms perform better for the Enron mail dataset compared to Cell Phone Calls. In addition, our NMI in datasets 1 and 3 is above 0.98 and the error ratio is below 2400. Finally, it can be said that our proposed algorithm is better than other algorithms in all evaluations.

## 5.4. Scalability Analysis

The evolutionary algorithms are time consuming because of the evaluation of repeated fitness function, especially for multi-objective methods, when the population size of the network



(a)  $g = 50$



(b)  $g = 100$

**Figure 14** The comparison of operation time of MOGWO-LP and DYNMOGA (in seconds) with the number of nodes.

is high. MOGWO-LP is an algorithm which can calculate the fitness function efficiently especially in large-scale networks.

In order to demonstrate the scalability of MOGWO-LP, dataset 1 ( $avgDegree = 16$ ,  $z = 5$ ,  $nC = 10\%$ ) was applied which increases the number of nodes  $n$  from 128, 256, 512, 1024, and 2048 to 4096. The population size ( $p$ ), which is the number of wolves, changes in the interval of [50, 100, 200], and the number generations ( $g$ ), which is the number of repetitions, is different at the scope of 50-100. The comparison of operation time of MOGWO-LP and DYNMOGA (in seconds) with the number of nodes is shown in Figure 14, using different combinations of population and the number of generations.

In Figure 13(a) and 13(b), the operation time of onetime step for the various mixture of  $p$  and  $g$  have been plotted. Based on the measurements, MOGWO-LP is less time consuming considerably compared to DYNMOGA when the number of nodes increases over time.

In fact, the runtime of MOGWO-LP grows roughly linearly with respect to the sum of nodes and the number of edges. The linearity of our runtime verifies the analysis of the proposed complexity representing a fine scalability feature.

## 5.5. Time Complexity

The time complexity of the MOGWO-LP algorithm is actually the time which is consumed in calculating the LP algorithm being

$O(m)$ , in which  $m$  shows the number of edges in the graph. The time complexity for the MOGWO algorithm is to find the best solution for  $O(p * m) + O(p)$ , and update the wolves which is  $O(p * n)$ , so that  $p$  indicates the number of wolves and  $n$  is the number of nodes. Further, the time complexity for the modularity calculation is  $O(m)$  and for the NMI is  $O(n)$  [49].

In the presented algorithm, the time complexity in the first phase is  $O(I(pm + p + pn + m))$ , and in the second phase  $IT(pm_c + p + pn' + m_c + n)$ , where,  $I$  indicates the number of algorithm repetitions,  $m_c$  is considered as the average number of edges in each community,  $n'$  means the number of active nodes and  $T$  is regarded as the number of snapshots. Finally, the time complexity of the MOGWO-LP algorithm is:

$$I(pm + p + pn + m) + IT(pm_c + p + pn' + m_c + n) \Rightarrow O(Ip(m + n) + ITp(m_c + n' + n))$$

The temporal complexity of the other algorithms is as follows. Since the parameter  $T$  is a constant number and has a small value, it can be omitted in the temporal complexity. For this reason, in the DYNMOGA and L-DMGA algorithms, the authors omitted  $T$ .

DYNMOGA:  $O((gp \log p) \times (n \log n + m))$  [22] //  $p$  = population size and  $g$  = iterations

L-DMGA:  $O(gp \cdot \log(p) \times (n + m))$  [20]

ALPA:  $O(TIm_c)$  [21]

According to the above temporal complexities, ALPA is the best, then our proposed algorithm has better temporal complexity and then L-DMGA and DYNMOGA, respectively.

## 6. CONCLUSION

An improved multi-objective evolutionary algorithm was presented in this article, by integrating the GWO algorithm and improved LP algorithm for communities' detection in dynamic networks.

This proposed approach has two phases. In the first snapshot, the MOGWO-LP algorithm is applied to all nodes in the network to detect the communities. In the second phase, in the next snapshots, the proposed algorithm of detecting communities is applied on those parts of the network, which only includes active nodes. In each snapshot, the best solution of the Pareto Front is chosen with the maximum modularity. Of course, in the second phase, in a very small number of next snapshots, MOGWO-LP is applied to the entire network to have a better performance.

The tests on two real and artificial datasets indicated that MOGWO-LP algorithm produces better results than other algorithms in quality and detection speed terms. The suggested procedure is appropriate for networks with different sizes, even large networks and with a dynamic structure.

## CONFLICTS OF INTEREST

The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report.

## AUTHORS' CONTRIBUTIONS

Fatemeh Besharatnia, Alireza Talebpur, Sadeh Aliakbary contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. Alireza Talebpur supervised the project.

## Funding Statements

The authors received no financial support for the research, authorship, and/or publication of this article.

## ACKNOWLEDGMENTS

We are grateful to Hamidreza Mahdiani, Farshad Safaei, and the anonymous reviewers for their careful comments.

## REFERENCES

- [1] M. Takaffoli, R. Rabbany, O.R. Zaiane, Incremental local community identification in dynamic social networks, in Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Niagara Ontario Canada, Canada, 2013, pp. 90–94.
- [2] M.E. Samie, A. Hamzeh, Community detection in dynamic social networks: a local evolutionary approach, *J. Inf. Sci.* 43 (2016), 615–634.
- [3] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, *Proc. Natl. Acad. Sci.* 99 (2002), 7821–7826.
- [4] L. Danon, A. Dazguilera, J. Duch, A. Arenas, Comparing community structure identification, *J. Stat. Mech. Theory Exper.* 09 (2005), pp. 09008.
- [5] A. Lancichinetti, F. Radicchi, J.J. Ramasco, S. Fortunato, Finding statistically significant communities in networks, *PLOS ONE.* 6 (2011), 1–18.
- [6] R. Aldecoa, I. Maroán, Surprise maximization reveals the community structure of complex networks, *Sci. Rep.* 3 (2013), 1060.
- [7] S. Gregory, Finding overlapping communities in networks by label propagation, *N. J. Phys.* 12 (2010), 103018.
- [8] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in largescale networks, *Phys. Rev. E.* 76 (2007), 036106.
- [9] J. Xie, B.K. Szymanski, LabelRank: a stabilized label propagation algorithm for community detection in networks, in 2013 IEEE 2nd Network Science Workshop (NSW), West Point, NY, USA, 2013, pp. 138–143.
- [10] A. Singh, M.D. Humphries, Finding communities in sparse networks, *Sci. Rep.* 5 (2015), 8828.
- [11] M. Rosvall, C.T. Bergstrom, An information-theoretic framework for resolving community structure in complex networks, *Proc. Natl. Acad. Sci.* 104 (2007), 7327–7331.
- [12] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, *Proc. Natl. Acad. Sci.* 105 (2008), 1118–1123.
- [13] J.P. Bagrow, E.M. Bollt, Local method for detecting communities, *Phys. Rev. E.* 72 (2005), 046108.

- [14] Y.R. Lin, Y. Chi, S. Zhu, H. Sundaram, B.L. Tseng, Facetnet: a framework for analyzing communities and their evolutions in dynamic networks, in The 17th International World Wide Web Conference (WWW08), Beijing, China, 2008, pp. 685–694.
- [15] M.E.J. Newman, Modularity and community structure in networks, *Proc. Natl. Acad. Sci.* 103 (2006), 8577–8582.
- [16] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, *J. Stat. Mech. Theory Exper.* 2008 (2008), P10008.
- [17] Z. Masdarolomoor, R. Azmi, S. Aliakbary, N. Riahi, Finding community structure in complex networks using parallel approach, in 2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing, IEEE, Melbourne, VIC, Australia, 2011, pp. 474–479.
- [18] Y. Wang, B. Wu, N. Du, Community evolution of social network: feature, algorithm and model, arXiv:0804.4356, 2008.
- [19] M. Takaffoli, F. Sangi, J. Fagnan, O. Zaiane, MODEC — Modeling and Detecting Evolutions of Communities, International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 2011, pp. 30–41.
- [20] P.G. Sun, L. Gao, Y. Yang, Maximizing modularity intensity for community partition and evolution, *Inf. Sci.* 236 (2013), 83–92.
- [21] X. Niu, W. Si, C.Q. Wu, A label-based evolutionary computing approach to dynamic community detection, *Comput. Commun.* 108 (2017), 110–122.
- [22] J. Han, W. Li, L. Zhao, Z. Su, Y. Zou, W. Deng, Community detection in dynamic networks via adaptive label propagation, *PLOS ONE.* 12 (2017), e0188655.
- [23] F. Folino, C. Pizzuti, An evolutionary multiobjective approach for community discovery in dynamic networks, *IEEE Trans. Knowl. Data Eng.* 26 (2014), 1838–1852.
- [24] Y.R. Lin, Y. Chi, S. Zhu, H. Sundaram, B.L. Tseng, Analyzing communities and their evolutions in dynamic social networks, *ACM Trans. Knowl. Discov. Data.* 3 (2009), 307–308.
- [25] N. Dakiche, F.B.S. Tayeb, Y. Slimani, K. Benatchba, Community evolution prediction in dynamic social networks using community features' change rates, The 34th ACM/SIGAPP Symposium on Applied Computing, New York, NY, United States, 2019. 2078–2085
- [26] Z. Zhao, Ch. Li, X. Zhang, F. Chiclana, An incremental method to detect communities in dynamic evolving social networks, *Knowl-Based Syst.* 163 (2018), 404–415.
- [27] Y. Zhang, B. Wu, N. Ning, C. Song, J. Lv, Dynamic topical community detection in social network: a generative model approach, *IEEE Access.* 7 (2019), 74528–74541.
- [28] F. Liu, J. Wu, C. Zhou, J. Yang, Evolutionary community detection in dynamic social networks, in International Joint Conference on Neural Networks, Budapest, Hungary, 2019.
- [29] X. Li, B. Wu, Q. Guo, X. Zeng, C.H. Shi, Dynamic community detection algorithm based on incremental identification, in IEEE 15th International Conference on Data Mining Workshops, Atlantic City, NJ, USA, 2015.
- [30] S. Pérez-Peló, J. Sánchez-Oro, R. Martín-Santamaría, A. Duarte, On the analysis of the influence of the evaluation metric in community detection over social networks, *Multidiscip. Digit. Publ. Inst.* 8 (2019), 23.
- [31] H. Cheraghchi, A. Zakerolhosseini, S. Bagheri Shouraki, E. Homayounvala, A novel granular approach for detecting dynamic online communities in social network, *Soft Comput.* 23 (2019), 10339–10360.
- [32] P. Wu, P. Li, Multi-objective community detection method by integrating users' behavior attributes, *Neurocomputing.* 210 (2016), 13–25.
- [33] L. Wang, Y.Q. Liang, Q.J. Tain, et al., A community detection method based on multi-objective optimization method, *Appl. Mech. Mater.* 571–572 (2014), 177–182.
- [34] D. Zhuang, J.M. Chang, M. Li, DynaMo: dynamic community detection by incrementally maximizing modularity, *IEEE Trans. Knowl. Data Eng.* 33 (2019), 1934–1945.
- [35] J. Choi, G. Choi, Community detection based on social influence in large scale networks, in Workshops of the International Conference on Advanced Information Networking and Applications, Springer, Cham, 2020, pp. 122–137.
- [36] X. Han, D. Chen, H. Yang, A semantic community detection algorithm based on quantizing progress, *Complexity.* 2019 (2019), 1–13.
- [37] F. Liu, J. Wu, S. Xue, C. Zhou, J. Yang, Q. Sheng, Detecting the evolving community structure in dynamic social networks, *World Wide Web.* 23 (2020), 715–733.
- [38] E. Jokar, M. Mosleh, Community detection in social networks based on improved label propagation algorithm and balanced link density, *Phys. Lett. A.* 383 (2018), 718–727.
- [39] Y. Chi, X. Song, D. Zhou, et al., On evolutionary spectral clustering, *ACM Trans. Knowl. Discov. Data.* 3 (2009), 17.
- [40] M. Ehrgott, *Multicriteria Optimization*, Lecture Notes in Economics and Mathematical Systems, vol. 491, Springer, Berlin, Heidelberg, Germany, 2000.
- [41] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* 69 (2004), 026113.
- [42] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014), 46–61.
- [43] R. Hosseini Mazaher, R. Azmi, Improving label propagation algorithm for detecting communities on social networks, in 23rd Iranian Electrical Engineering Conference, Sharif University of Technology, Tehran, Iran, 2015.
- [44] F. Folino, C. Pizzuti, A multiobjective and evolutionary clustering method for dynamic networks, in International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Odense, Denmark, 2010, pp. 256–263.
- [45] M.S. Kim, J. Han, A particle-and-density based evolutionary clustering method for dynamic networks, *Proc. VLDB Endowment.* 2 (2009), 622–633.
- [46] L. Andrea, F. Santo, R. Filippo, Benchmark graphs for testing community detection algorithms, *Phys. Rev. E.* 78 (2008), 561–570.
- [47] IEEE VAST 2008 CHALLENGE, <http://www.cs.umd.edu/hcil/VASTchallenge08>, last accessed: April 23, 2021.
- [48] Enron Email Dataset, <http://www.cs.cmu.edu/~enron>, last accessed: April 23, 2021.
- [49] C.T. Liu, B.G. Hu, Mutual information based on renyi's entropy feature selection, in IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2009), Shanghai, China, 2009, pp. 816–820.