NECAMO - New Evolutionary Clustering Algorithm using Multiobjective Optimization

Md. Shiplu Hawlader [1], Kaynat Quayyum [1] and Saifuddin Md. Tareeq [1*]

[1]Department of Computer Science and Engineering

University of Dhaka, Dhaka 1000, Bangladesh

shiplu.cse.du@gmail.com , *smtareeq@cse.univdhaka.edu*

Abstract

Evolutionary clustering in dynamic networks is the process of generating a sequence of clusters to capture the true nature of evolution in interactions among individuals. There are several different approaches to solve the evolutionary clustering problem and a relatively new one is the multiobjective optimization using genetic algorithms. In this paper, we have proposed a new evolutionary clustering algorithm using multiobjective optimization or NECAMO. Experimental results on real and synthetic data show that the proposed algorithm provides better accuracy compared to previous algorithms in terms of the two relevant parameters - accuracy of clustering with respect to current data and drift from the clusters found from previous steps.

1 Introduction

Individuals or objects are logically connected in networks, either densely, or sparsely. The dense group can be defined as a cluster or community. Clustering is the organization of data patterns into groups based on some measures of similarities or logical connections [10]. The study of real world dynamic networks is a growing scientific field that combines traditional social network analysis (SNA), link analysis (LA) and multi-agent systems (MAS) within network science and network theory. Most of the

[*]to whom correspondence should be addressed

complex dynamic networks can be modeled as graphs, where nodes represent individual objects and edges represent relation among those individuals. The dynamic behavior of the network can be imposed by keeping a set of graphs for a particular time window. Each static graph in the set corresponds to the network for a particular timestamp. The proximity between individuals can be modeled by putting weight in the edges. The uncertainty however can be established in the transformation of graphs over times. Dynamic network analysis leads to finding pattern in the clustering of a community. Hence, this type of analysis can be beneficial for numerous entities like social network developers, telecommunication operators or for detecting the group of individuals responsible for spamming or junk emails.

Many approaches have been proposed for the analysis and temporal evolution of dynamic networks in [1], [3], [12], [13], [14], [15], [18], [19], [20], [21], [22]. To catch the evolution of clusters in dynamic network with temporal data, some of these methods [3], [12], [15], [20] employ the concept of Chakrabarti *et. al.* in [2]. According to Folino *et. al.* [8] evolutionary clustering is the process of grouping data coming at different time steps to produce a sequence of clusters by introducing a framework called temporal smoothness. This framework assumes that abrupt changes of clustering in a short time period are not desirable, thus it smooth each community over time. Smoothness is defined by trading-off between two competing criteria. *Snapshot cost*, which means that the clustering should reflect as nearly as possible the current time steps coming data. And *temporal cost*, which means that each clustering should not change dramatically from the previous time steps.

In this paper we have introduced NECAMO algorithm which implements similar concept as [23], but with an improved fitness assignment scheme. In the following sections we will elaborate the proposed algorithm.

2 Problem Formulation

Let $\{1, \ldots, T\}$ be a finite set of time steps and $V = \{1, \ldots, |V|\}$ be a set of individuals or objects. A static network $\mathcal{N}^t$ at time t can be modeled as a graph $G^t = (V^t, E^t)$ where V^t is a set of objects, called nodes or vertices and E^t is a set of links, called edges, that connect two elements of V^t at time t. Thus G^t is the graph representing a snapshot of the network $\mathcal{N}^t$ at time t. An edge $(u^t, v^t) \in E^t$ if individual u and v have interacted at time t.

A dynamic network $\mathcal{N}$ is a sequence of static networks $\{\mathcal{N}^1, \ldots, \mathcal{N}^T\}$, where each network $\mathcal{N}^t$ is a snapshot of the network at timestamp t. A cluster in a static network $\mathcal{N}^t$ is a group of vertices consisting of a subset of V^t having a high density of edges inside the cluster and a lower density of edges outside of the cluster [8]. Let C^t is a sub-graph of G^t be a cluster at time t. A clustering $\mathcal{CR}^t = \{C_1^t, \dots, C_k^t\}$ of network $\mathcal{N}^t$ at time t is a partitioning of the graph G^t in groups of nodes such that for each couple of groups or communities C_i^t and $C_j^t \in \mathcal{CR}^t$ and $V_i^t \cap V_j^t = \emptyset$.

A multiobjective optimization problem deals with more than one objective functions that are to be minimized or maximized. These objectives can be conflicting, subject to certain constraints and often lead to choosing the best trade-off among them. Evolutionary clustering comprises of two conflicting objective functions to optimize.

2.1 Multiobjective Evolutionary Clustering

A multiobjective evolutionary clustering problem $(\Omega, \mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_h)$ for a static network $\mathcal{N}^t$ can be defined as $min\mathcal{F}_i(\mathcal{CR}^t)$, $i = 1, \dots, h$ where $\Omega = \{\mathcal{CR}_1^t, \dots, \mathcal{CR}_k^t\}$ is the set of feasible clusters of $\mathcal{N}^t$ at time stamp t, and $\mathcal{F} = \mathcal{F}_1, \dots, \mathcal{F}_h$ is a set of h single criterion functions [8]. Each $\mathcal{F}_i \colon \Omega \rightarrow R$ is a different objective function that determines the feasibility of the clustering obtained. $\mathcal{F}$ is a vector of conflicting objectives that must be simultaneously optimized, there is not one unique solution to the problem, but a set of solutions are found through the use of Pareto optimality theory [7].

Given two solution $\mathcal{CR}_1$ and $\mathcal{CR}_2 \in \Omega$, solution $\mathcal{CR}_1$ is said to dominate solution $\mathcal{CR}_2$, denoted as $\mathcal{CR}_1 \prec \mathcal{CR}_2$, if and only if ,

$$\forall i \colon \mathcal{F}_i(\mathcal{CR}_1) \leq \mathcal{F}_i(\mathcal{CR}_2) \wedge \exists i \;\; \mathcal{F}_i(\mathcal{CR}_1) < \mathcal{F}_i(\mathcal{CR}_2)$$

That is, all the objective function of $\mathcal{CR}_1$ should be better or equal to $\mathcal{CR}_2$ and at least one objective function is better than $\mathcal{CR}_2$. On the other hand, non-dominated solutions is one for which an upgrading in one objective need a ruin of another. These solutions are called Pareto-optimal. Formally the set of Pareto-optimal solutions Π is defined as

$$\Pi = \{\mathcal{CR} \in \Omega \colon \forall \mathcal{CR}' \in \Omega \, with \, \mathcal{CR}' \prec \mathcal{CR}\}$$

2.1.1 Snapshot cost

Snapshot Cost (SC) measures how well a cluster C^t represents the network $\mathcal{N}^t$ at time t. *Community Score* is very effective to detect snapshot cost of cluster [17]. Let $\mathcal{CR}^t = \{C_1^t, \dots, C_k^t\}$ be a clustering of a network $\mathcal{N}^t$ of graph G^t at time t. The *community score* of $\mathcal{CR}^t$ is defined as follows,

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

where

$$score(C_i^t) = \frac{\sum_{j \in C^t} (\mu_j)^2}{|C^t|} \times \sum_{i,j \in C^t} A_{ij}^t \tag{2}$$

In equation (2),second term represents number of edge connecting nodes inside the community C^t. The first term computes the square mean of,

$$\mu_i = \frac{1}{|C^t|} \sum_{j \in C^t} A_{ij}^t \tag{3}$$

where A_{ij}^t is the adjacency matrix of 1 and 0. In equation (2) and (3) μ_i how tightly a node i is connected in the cluster C^t by fraction of edge connection inside C^t. Thus according to equation (1), (2) and (3) the objective function *community score* is considering the number of interconnection inside the community and also outside the community. More the *community score*, more the cluster represents the current network $\mathcal{N}^t$.

2.1.2 Temporal cost

The algorithm should maintain the similarity between the current clusters to the previous time steps clusters. Normalized Mutual Information (NMI), well known entropy measure in information theory, can be used to measure temporal cost. Let two clusters $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}$ of a network. let M be the confusion matrix whose cell M_{ij} represents how many nodes in cluster $A_i \in A$ that are also in the cluster $B_j \in B$. The Normalized Mutual Information (NMI) is defined as,

$$NMI(A,B) = \frac{-2 \sum_{i=1}^{n} \sum_{j=1}^{m} M_{ij} \log \frac{M_{ij} N}{M_{i.} M_{.j}}}{\sum_{i=1}^{n} M_{i.} \log(M_{i.}/N) + \sum_{j=1}^{m} M_{.j} \log(M_{.j}/N)} \tag{4}$$

where n and m are number of groups of the clustering A and B respectively, $M_{i.}$ and $M_{.j}$ are sum of the $i-th$ row and sum of the $j-th$ column of confusion matrix M respectively and N is the number of nodes in the network. NMI is a value between 0 and 1. If clustering A and B are same, that is $A = B$, then $NMI(A,B) = 1$ and if A and B are completely different then $NMI(A,B) = 0$. Thus the objective function for this problem $NMI(\mathcal{CR}^t, \mathcal{CR}^{t-1})$ are to be maximized.

2.2 Multiobjective Genetic Algorithm

Genetic Algorithms (GA) can provide very efficient solutions for multiobjective optimizations. A generic single objective GA can be modified to find a set of multiple non-dominated solutions in clustering method. A multi-objective GA should have a good fitness function, should preserve the diversity in solutions and a good multiobjective genetic algorithm should maintain elitism.

2.2.1 Fitness Function

There are several ways of calculating fitness function:

- Weighted Sum Approach, in which each objective function is multiplied with a weight. When multiple solutions are required, the same objective functions are multiplied with different weight combinations.
- Altering objective function, where the population P is randomly divided into k equal sized sub-population $P_1, P_2, ..., P_k$ and each sub population is assigned a fitness value based on objective function z.
- Pareto Ranking Approaches, where, the individuals are ranked according to dominance rule.

2.2.2 Diversity Measure

Maintaining diversity in population is important because we need those solutions that are distributed over the problem domain uniformly. Following are the issues for diversity measurement.

Fitness Sharing: The idea of this process is as follows; at first, the euclidean distance between each individual pairs are calculated by the given formula:

$$dz(u,v) = \sqrt{\sum_{k=1}^{K} \left(\frac{z_k(u) - z_k(v)}{z_k^{max} - z_k^{min}} \right)^2} \tag{5}$$

where u and v are two solution pairs, z_k^{max} and z_k^{min} are maximum and minimum value of the objective function z_k. After that, the niche count is calculated. Decision variabls can also be used to calculate the distance between two solutions.

Crowding Distance: This method can be used as a diversity measure. Here, the individuals are ranked and then crowding distance is calculated as follows:

$$cd_k(v_{[i,k]}) = \frac{z_k(v_{[i+1,k]}) - z_k(v_{[i-1,k]})}{z_k^{max} - z_k^{min}} \tag{6}$$

Cell based density: In this technique, the problem domain is divided into k equal sized cells, and each individual belongs to one cell.

2.2.3 Elitism

All the non-dominated solutions are elite solution. Normally there could be large number of elite solutions in multiobjective GA and it is impossible to save all the elite solutions. Recent multiobjective GAs like [6], [5] and [4], implements elitism found far better accuracy with respect to non-elitist multiobjective GAs like in [11].

3 Algorithm

It takes a dynamic network $\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_T\}$, the sequence of graphs $G = \{G_1, G_2, \ldots, G_T\}$ and the number of timestamps T as input and gives a clustering of each network $\mathcal{N}_i$ of $\mathcal{N}$ as output.

3.1 Initialization

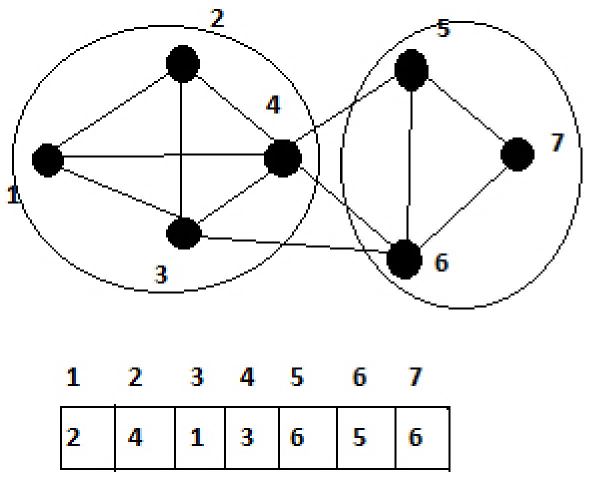
For the first timestamp of first input network there is no temporal relation with the previous network. The clustering algorithm is applied only in snapshot cost.

3.2 Population from Graph

As a first step in each timestamp from 2nd timestamp to T, it creates a population of random individuals. Each individual is a vector of length equal to number of nodes in the graph G^t.

3.3 Decoding

As each individual gene is working as an adjacency list, if a node in u of graph is reachable from v by maintaining the edges in the individual, then u and v is in same cluster of component.



1	2	3	4	5	6	7
2	4	1	3	6	5	6

Figure 1: Network of 7 nodes clustered into {1,2,3,4} and {5,6,7} and their genetic reprsentation

3.4 Evaluation

In this algorithm the evaluation phase consist of calculating community score, which defines how better is the current clustering with respect to the given network and normalized mutual information (NMI), which defines the clustering fluctuation from the previous timestamp.

3.5 Assign Rank

Each individual of the population and archive is given a rank value, the smaller the better. After giving each individuals a rank value, sort the individuals according to the ascending rank.

$$r(v) = \sum_{v \prec u} s(u) \tag{7}$$

3.6 Fitness Function

To remain the population diverse, we are using distance of $k-th$ nearest neighbor. The fitness value of each individual is the sum of its non-dominated rank and the inverse of the distance of $k-th$ nearest neighbors distance.

$$f(v) = r(v) + (\sigma_v^k + 1)^{-1} \tag{8}$$

where σ_v^k is the distance between individual v and its $k-th$ nearest neighbor.

Parent1:	4	3	2	2	6	5	6
Parent2:	3	3	1	5	4	7	6
Mask :	0	1	1	0	0	1	1
Offspring:	4	3	1	2	6	7	6

Table 1: Example of Uniform Crossover

3.7 Population Selection

From the total individuals of population and archive population size individuals are selected as new population. From the rank 0 to the highest rank, all the individuals are added if number of population of this rank is not exceeding the current population size. If it is exceeding, then some individuals are truncated according to the value of each individuals.

3.8 Mating Pool Creation

A mating pool is created of pool size from the new population to apply the genetic variation operators. To choose the mating pool, binary tournament with replacement has been used in this algorithm.

3.9 Genetic Variation Operators

Genetic operators are used to create offspring from parent or mating pool.

3.9.1 Crossover

In this algorithm we are using uniform crossover. A random bit vector of length of number of the node in the current graph is created. If $i-th$ bit is 0 then the value of the $i-th$ gene comes from the first parent otherwise it comes from the $i-th$ gene of second parent. As each of the parents holding true adjacency information, the offspring will also hold it.

3.9.2 Mutation

To mutate and create a offspring, some position of the of the individuals are chosen randomly and changed to other values. But the value should be one of its neighbors in the current graph.

3.10 Archive

After variation operators over the offspring becomes the new population and the old populations are saved to the archive. To fit the archive size, the truncation mechanism used here also.

Algorithm 1 Algorithm: New Clustering Algorithm using Multiobjective Optimization (NECAMO)

```
1:  procedure NECAMO($\mathcal{N} = \{\mathcal{N}^1, \ldots, \mathcal{N}^T\}$ a dynamic network of graph $G = \{G^1, \ldots, G^T\}$)
2:      Generate initial cluster $\mathcal{CR}_1 = \{C_1^1, \ldots, C_k^1\}$ of thenetwork $\mathcal{N}^1$ by optimizing only ($\mathcal{SC}$)
3:      let $t = 2$
4:      while $t \leq T$ do
5:          Create initial population of random individual $P_0$ and set $E_0 = \emptyset, i = 0$
6:          while $\Delta f(v) \geq 10^{-5}$ do
7:              Select individuals from $E_i$ for mating pool using binary tournament with replacement
8:              Apply crossover and mutation to create $N_P$ offspring solutions. Copy offspring to $P_i$
9:              Decode each individual of $P_i \cup E_i$
10:             Evaluate each individual of $P_i \cup E_i$ and assign fitness value using equation 8
11:             Copy all non-dominated solutions to $E_{i+1}$
12:             if $|E_{i+1}| > N_E$ then
13:                 truncate $|E_{i+1}| - N_E$ solutions according to maximum density value
14:             else
15:                 copy best $N_E - |E_{i+1}|$ dominated solutions according to their fitness value
16:             end if
17:             Set $i = i + 1$
18:         end while
19:         $C_t = E$ with highest modularity
20:         Set $t = t + 1$
21:     end while
22:     Return the clustering set $C$ with best cluster.
23: end procedure
```

4 Performance Analysis

NMI and modularity are calculated as a measure of performance. Modularity defines the goodness of a clustering. Good clustering, which have high value of modularity, are those in which there are dense internal connections between the nodes within cluster but only sparse connections between different clusters. In this algorithm we use the modularity, introduced by Girvan and Newman [16]. Let k be the number of clusters found inside a network, the modularity Q is defined as,

$$Q = \sum_{C=1}^{k} \left[\frac{l_C}{m} - \left(\frac{d_C}{2m} \right)^2 \right] \tag{9}$$

where m is the number of edges of the network, l_C is the total number of edges joining vertices inside the cluster C, and d_C is the sum of the degrees of the nodes of cluster C.

5 Experimental Result

In this section we will discuss the experimental result and its effectiveness and compare the result with one of the recent the algorithm namely Dyn-Moga proposed by Folino *et. al.* in [8]. For this purpose we used synthetic data as well as real life data. In both cases, our proposed algorithm successfully detects network structure and is very competitive with respect to other algorithms. Some standard parameter for the genetic algorithm are crossover probability 80%, mutation probability 20% and binary tournament for pool selection [9]. The population size was 200, archive size of elitism was 200 and number of generations 300.

5.1 Synthetic Data

We have created two types of synthetic data, in one number of different cluster in each timestamp is fixed and in other one number of cluster changes with timestamp. For the first type of network, we fixed number of node 128 and it is divided into 4 groups each of 32 nodes. Each node has average degree of 16. As in [8] we have fixed two parameter Z_{in} and Z_{out}. Each node has Z_{in} number of connection to the internal nodes in which cluster it is situated and Z_{out} number of connection to the outside of the cluster. These connections are created randomly. Total ten network data has been created for ten timestamp.

For the second type of network, we have changed the number of clusters in a timestamp to test whether the algorithm can detect the change in network structure. Total number of node this time is

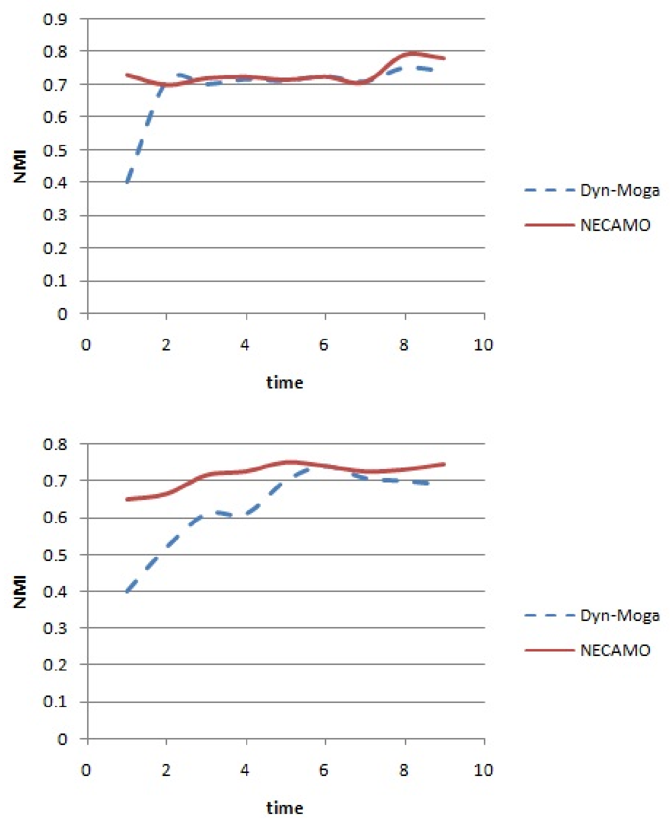


Figure 2: Normalized Mutual Information with fixed clustering when $Z_{out} = 3$ (top) and $Z_{out} = 5$ (bottom).

256 and for the first timestamp it is divided into 4 clusters, each having 64 nodes. Ten consecutive networks are generated by choosing eight nodes from each community and generating new cluster with these 32 nodes.

Figure: 2 shows the obtained result for the synthetic data with fixed number of clustering. In Figure: 2.1 shows that NMI value with $Z_{out} = 3$ for [8] and our proposed algorithm are close but in Figure: 2.2 our algorithm outperforms the [8] algorithm with $Z_{out} = 5$. According to Figure: 3 our algorithm is much stable in historical aspect with respect to [8]. It also have better modularity in both case of synthetic data compared to [8] according to Figure: 4.

5.2 Real-life Data

As real life data set we have chosen the Football data[1] from the United States college football. The football data is from NCAA Football Division 1-A games. Nodes in the graph represent the team and

[1]http://www.jhowell.net/cf/scores/scoresindex.htm

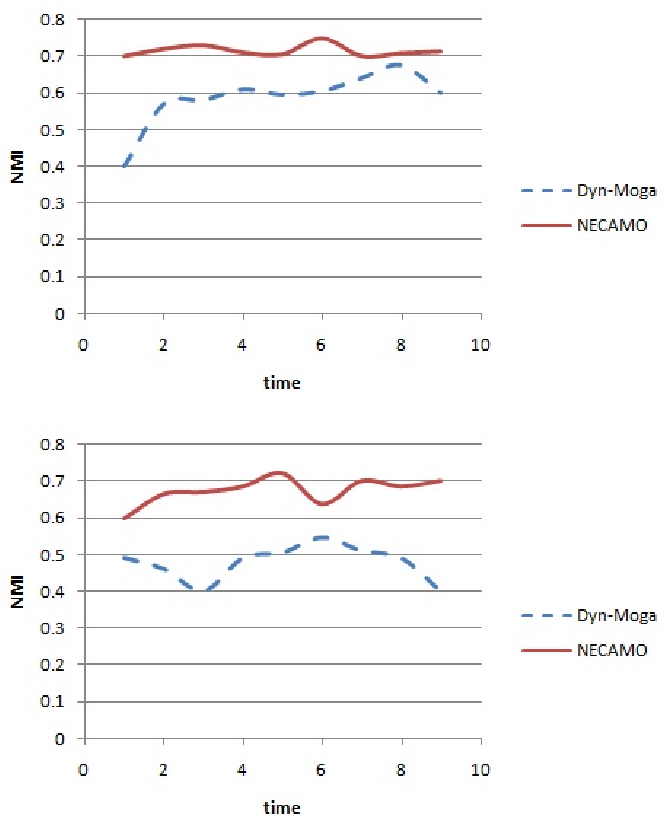


Figure 3: Normalized Mutual Information with variable clustering when $Z_{out} = 3$ (top) and $Z_{out} = 5$ (bottom).

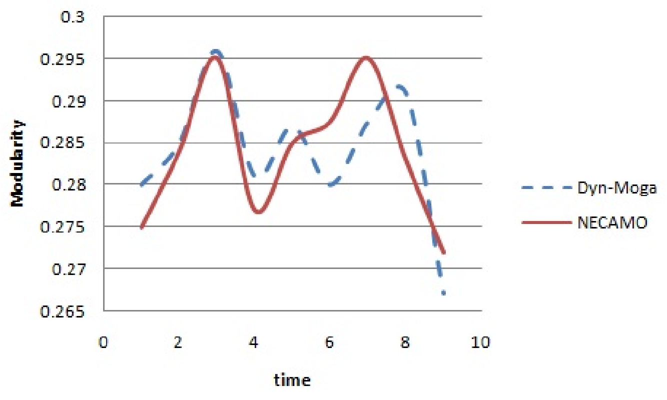


Figure 4: Modularity values obtain with fixed clustering and $Z_{out} = 3$

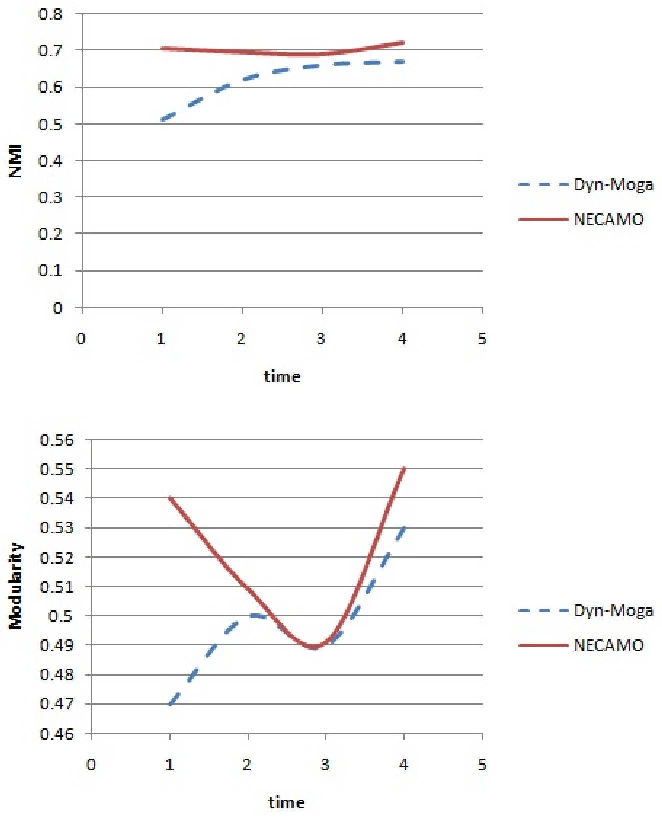


Figure 5: Result obtained with real-life data, NMI and Modularity

edges represent the regular season games between two teams they connect. The teams are divided into conferences and they tend to play between members of the same conference. Thus the team cluster is assumed to be the conference. We selected 5 years data from 2006 to 2010 as the 5 timestamps network. Total nodes are 180, 197, 204, 208 and 216 respectively. The Figure: 5 shows that, our proposed algorithm outperforms the [8] algorithm in case of normalized mutual information (Figure: 5.1) and Modularity (Figure: 5.2).
After these analysis we are claiming that, our proposed algorithm will perform better and give more accurate result with compared to previous algorithms on evolutionary clustering.

6 Conclusion

The need for accuracy and performance in the field of evolutionary algorithm are the reason for our algorithm to implement. In this paper we have provided a multiobjective genetic algorithm. The algorithm gives best solution which is a trade-off between two objectives current data and displacement from history, at each steps of the algorithm. Experimental result with real-life data as well as synthetic data shows that it smooths the clustering from one step to next and gives better result with compared to previous algorithms.

REFERENCES

[1] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framwork for characterizing the evolutionary behavior of interaction graphs. In *ACM Transaction on Knowledge Discovery from Data*, volume 3 of *16*, pages 554–560. ACM, 2006.

[2] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, editors, *KDD*, pages 554–560. ACM, 2006.

[3] Y. Chi, D. Zhou, X. Song, K. Hino, and B. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proc. of the International Conference on Knowledge Discovery and Data Mining, (KDD'07)*, pages 153–162, 2007.

[4] V. V. DA and L. GB. Multiobjective evolutionary algorithms: analyzing the state-of-art. pages 125–147, 2000.

[5] K. Deb. Multi-objective optimization using evolutionary algorithms. NewYork,USA, 2001. Willey.

[6] Z. E and T. L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. pages 257–271, 1999.

[7] M. Ehrgott. Multicriteria Optimization. Berlin,Germany, 2005. Springer.

[8] F. Folino and C. Pizzuti. A multiobjective and evolutionary clustering method for dynamic networks. Institute for High Performance Computing and Networking (ICAR) and Italian National Research Council, 2010.

[9] D. E. Golberg and K. Deb. A Comparative Analysis of Selection Scheme used in GAs. In *G. Rawlins (Ed.), Foundations of GAs (FOGA), San Mateo, CA: Morgan Kaufmann*, pages 69–93, 1991.

[10] A. K. Jain, M. Murty, and P. Flynn. Data Clustering . In *ACM Comuting Surveys*, volume 31, pages 264–296, 1999.

[11] M. T. Jensen. Reducing the run-time complexity of multiobjective eas: the nsga-ii and other algorithms. pages 503–515, 2003.

[12] M. Kim and J. Han. A particle-and-density based evolutionary clustering method for dynamic networks. In *Proc. of the International Conference on Very Large Databases,(VLDB'09)*, 2009.

[13] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Proc. of the International Conference on Knowledge Discovery and Data Mining, (KDD'06)*, pages 611–717, 2006.

[14] L. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. of the International Conference on Knowledge Discovery and Data Mining, (KDD'05)*, pages 177–187, 2005.

[15] Y.-R. Lin, S. Zhu, H. Sundaram, and B. L. Tseng. Facetnet: A framework for analyzing communities and their evolutions in dynamic networks. In *Proc. of the International World Wide Web Conference (WWW'08)*, pages 685–694, 2008.

[16] M. Newman and M. Girvan. Finding and evaluating community structure in networks. In *Physical Review E*, volume 69, pages 6113–6127, 2004.

[17] C. Pizzuti. Ga-net: a genetic algorithm for community detection in social networks. In *Proc. of 10th International Conference on Parallel Problem Solving from Nature*, pages 1081–1090, 2008.

[18] C. Pizzuti. A multi-objective genetic algorithm for community detection in networks. In *Proc. of the 121st IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2009)*, pages 379–386, 2009.

[19] J. Sun, C. Faloutsos, S. Papadimitriou, and P. Yu. Graphscope:parameter-free of large time evolving-graphs. In *Proc. International Conference on Knowledge Discovery and Data Mining (KDD05)*, pages 687–696, 2005.

[20] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *Proc. International Conference on Knowledge Discovery and Data Mining (KDD07)*, pages 677–685, 2007.

[21] T. Xu, Z. Zhang, P. S. Yu, and B. Long. Dirichlet process based evolutionary clustering. In *Proc. of the 8th IEEE International Conference on Data Mining (ICDM08)*, pages 648–657, 2008.

[22] T. Xu, Z. Zhang, P. S. Yu, and B. Long. Evolutionary clustering by hierarchical dirichlet process with hidden markov state. In *Proc. of the 8th IEEE International Conference on Data Mining (ICDM08)*, pages 658–667, 2008.

[23] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Zurich, Switzerland, 2001.