NECAMO - New Evolutionary Clustering Algorithm usingMultiobjective 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*

E-mail: shiplu@cse.univdhaka.edu and 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 [9]. 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 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], [20], [21], [22], [23], [24]. To catch the evolution of clusters in dynamic network with temporal data, some of these methods [3], [12], [15], [22] employ the concept of Chakrabarti*et. al.* in [2]. These methods 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 proposed NECAMO algorithm which implements similar concept as [6], but with an improved fitness assignment scheme. The proposed algorithm performs better in terms of accuracy with respect to previous algorithms. The result is supported by experimentation with synthetic and real data. In the following sections we will elaborate the proposed algorithm.

2. Related Work

Several genetic algorithms provided efficient solutions for multiobjective optimizations. Some well-known multiobjective genetic algorithms are vector evaluated GA [10], Niched Pareto Genetic Algorithm [8], Weight Based Genetic Algorithm [18], Non-dominated Sorting Genetic Algorithm (NSGA)[16], Strength Pareto Evolutionary Algorithm (SPEA) [4] and more. General problems with all these multi-objective evolutionary algorithms are that computational complexity is $O(MN^3)$ (where M is the number of objectives and N is the population size), their non-elitism approach and the need to specify a sharing parameter.

In Folino et al. [6] used NSGA2 [11] for detecting communities in dynamic networks. NSGA2 is the improved version of NSGA [16] which alleviates all the above three problems by introducing a new selection operator which creates a mating pool by combining the parent and offspring populations and used crowding distance for diversity preservation.

In our proposed algorithm, we have used SPEA2 [25]. SPEA2 is an improved algorithm of SPEA with respect to fitness and diversity measurement. SPEA2 used an improved fitness assignment scheme, which takes into account how many individuals dominate and dominated by a single individual. SPEA2 used k-th nearest neighbor to preserve diversity of solutions and gives better quality

population in short period of time than NSGA2. We have used SPEA2 to predict different cluster of network in different time stamp. In next sections we will show, our proposed algorithm gives better result with respect to [6].

3. Problem Formulation

Let $\{1,...,T\}$ be a finite set of time steps and $V = \{1,...,|V|\}$ 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 N is a sequence of static networks $\{\mathcal{N}^1,...,\mathcal{N}^t\}$, 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 [6]. Let C^t is a sub-graph of G^t be a cluster at time t. A clustering $CR^t = \{C_1^t,..., C_k^t\}$ 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 $C_j^t \in CR^t$ 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.

3.1 Multiobjective Evolutionary Clustering

A multiobjective evolutionary clustering problem $(\Omega, \mathcal{F}_1, \mathcal{F}_2,..., \mathcal{F}_h)$ for a static network $\mathcal{N}^t$ can be defined as $min\mathcal{F}_i(CR^t)$, $i= 1,...,h$ where $\Omega = \{ CR_1^t,..., CR_k^t\}$ is the set of feasible clusters of $\mathcal{N}^t$ at time stamp t, and $\mathcal{F} = \mathcal{F}_1,..., \mathcal{F}_h$ is a set of h single criterion functions [6]. Each $\mathcal{F}_i: \Omega \to R$ 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 [5].

Given two solution CR_1 and $CR_2 \in \Omega$, solution CR_1 is said to dominate solution CR_2, denoted as $CR_1 \prec CR_2$, if and only if,

$$\forall i : \mathcal{F}_i(CR_1) \leq \mathcal{F}_i(CR_2) \wedge \exists i \; \mathcal{F}_i(CR_1) < \mathcal{F}_i(CR_2)$$

That is, all the objective function of CR_1 should be better or equal to CR_2 and at least one objective function is better than 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

$$\Pi = \{CR \in \Omega : \forall CR' \in \Omega \text{ with } CR' \prec CR\}$$

3.1.1 Snapshot cost

Snapshot Cost (SC) measures how well a cluster C represents the network $\mathcal{N}^t$ at time t. Community Score is very effective to detect snapshot cost of cluster [19]. Let $CR^t = \{C_1^t,..., C_k^t\}$ be a clustering of a network $\mathcal{N}^t$ of graph G^t at time t. The community score of CR^t is defined as follows,

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

where,

$$score(C_i^t) = \frac{\sum_{j \in c^t}(\mu_i)^2}{|C^t|} \times \sum_{ij \in c^t} A_{ij}^t \qquad (2)$$

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

$$\mu_i = \frac{1}{|C^t|} \sum_{j \in c^t} A_{ij}^t \qquad (3)$$

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 $\mathcal{N}^t$.

3.1.2 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 $A = \{A_1,...,A_n\}$ and $B = \{B_1,...,B_m\}$ 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,

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

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 $NMI(CR^t, CR^{t-1})$ are to be maximized.

3.2 Multiobjective Genetic Algorithm

Genetic Algorithms (*GA*) can provide very efficient solutions for multi-objective 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 and should maintain elitism.
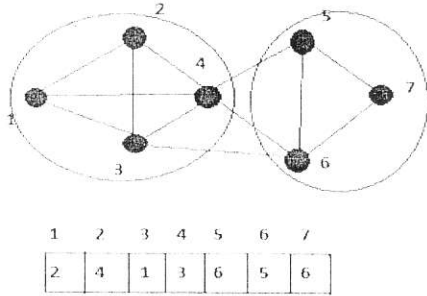

Fig 1: Network of 7 nodes clustered into {1,2,3,4} and {5,6,7} and their genetic representation.

4. Proposed Algorithm

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

4.1 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.

4.2 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'.

4.3 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.

4.4 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.

4.5 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.

$$r(v) = \sum_{v<u} s(u) \qquad (5)$$

4.6 Fitness Function

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

$$f(v) = r(v) + (\sigma_v^k + 1)^{-1} \qquad (6)$$

Where σ_v^k is the distance between individual v and its kth nearest neighbor.

4.7 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.

4.8 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.

Table 1: Example of Uniform Crossover

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

4.9 Genetic Variation Operators

Genetic operators are used to create offspring from parent or mating pool. There are two widely used genetic variation operators namely crossover and mutation. In the following subsection a short description of these two operators is given.

4.9.1 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.

4.9.2 Mutation

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

4.10 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(𝒩 = {𝒩¹,..., 𝒩ᵗ} a dynamic network of graph G = {G¹,...,Gᵗ})
2:       Generate initial cluster 𝒞ℛ₁ = {𝒞₁¹,..., 𝒞ₖ¹} of the network 𝒩¹ by optimizing only (𝒮𝒞)
3:       let t = 2
4:       while t ≤ T do
5:              Create initial population of random individual P₀ and set E₀ = ∅, i = 0
6:                     while Δf(v) ≥ 10⁻⁵ do
7:                            Select individuals from Eᵢ for mating pool using binary tournament with replacement
8:                            Apply crossover and mutation to create N_P offspring solutions. Copy offspring to Pᵢ
9:                            Decode each individual of Pᵢ∪Eᵢ
10:                           Evaluate each individual of Pᵢ∪Eᵢ and assign fitness value using equation (6)
11:                           Copy all non-dominated solutions to Eᵢ₊₁
12:                           if |Eᵢ₊₁| > N_E then
13:                                  truncate |Eᵢ₊₁|- N_E solutions according to maximum density value
14:                           Else
15:                                  copy best N_E-|Eᵢ₊₁| dominated solutions according to their fitness value
16:                           end if
17:                           Set i = i+ 1
18:                     end while
19:              Cᵢ = E with highest modularity
20:              Set t = t + 1
21:       end while
22:       Return the clustering set C with best cluster.
23:   end procedure
end
```

5. 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 [17]. Let k be the number of clusters found inside a network, the modularity Q is defined as,

$$Q = \sum_{C=1}^{k} \left[\frac{l_C}{m} - \left(\frac{d_C}{2m} \right)^2 \right] \tag{7}$$

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.

6. 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 [6]. 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 [7]. The population size was 200, archive size of elitism was 200 and number of generations 300.

6.1 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 [6] 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 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 (top) shows that *NMI* value with $Z_{out} = 3$ for [6] and our proposed algorithm are close but in bottom our algorithm outperforms the [6] algorithm with $Z_{out} = 5$. According to Figure: 3 our algorithm is much stable in historical aspect with respect to [6]. It also have better modularity in both case of synthetic data compared to [6] according to Figure: 4.
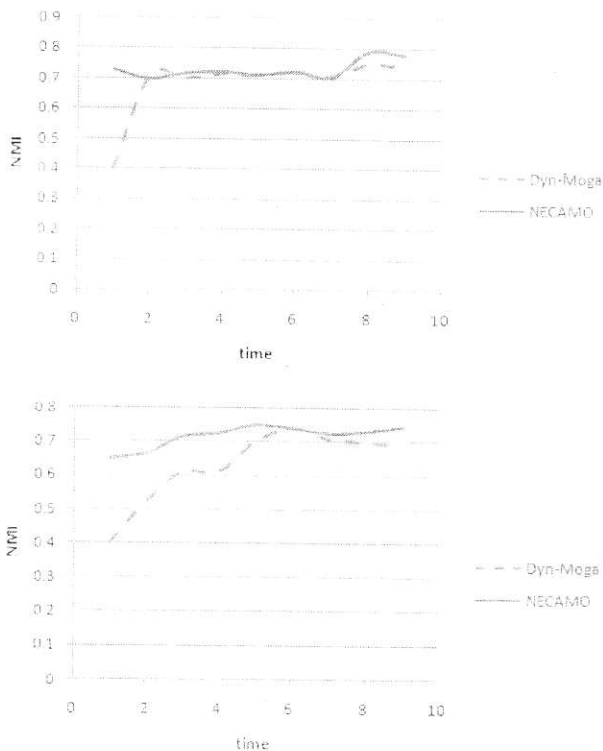

Fig. 2: Normalized Mutual Information with fixed clustering when $Z_{out} = 3$ (top) and $Z_{out} = 5$ (bottom).

6.2 Real-life Data

As real life data set we have chosen the Football data 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 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.
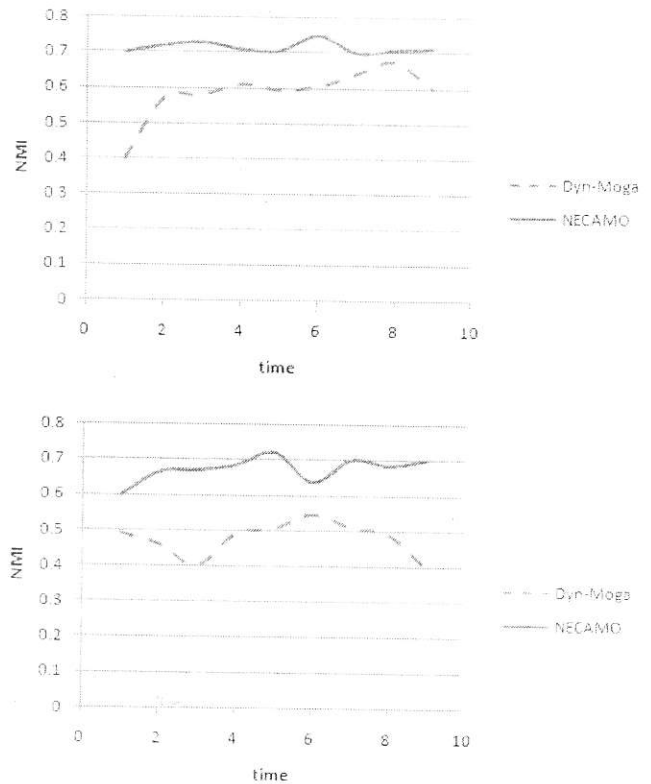

Fig. 3: Normalized Mutual Information with variable clustering when $Z_{out} = 3$ (top) and $Z_{out} = 5$ (bottom).
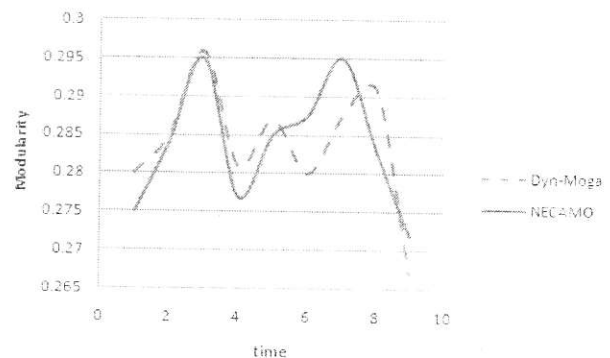

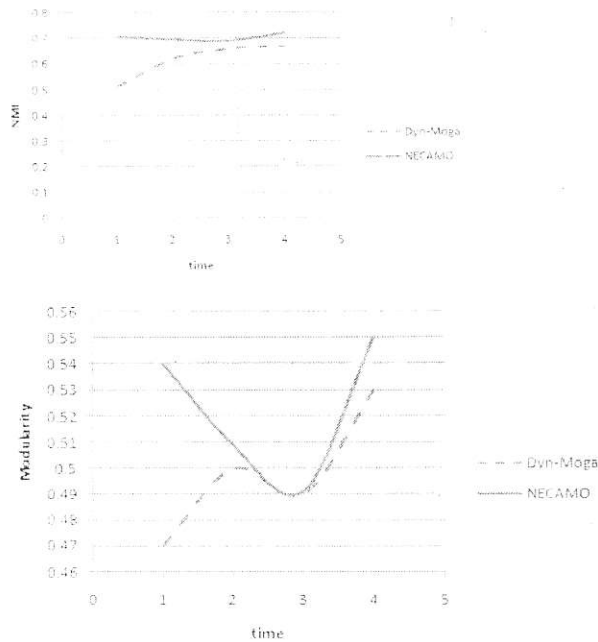Fig. 4: Modularity values obtain with fixed clustering and $Z_{out} = 3$

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

The Figure: 5 shows that, our proposed algorithm outperforms the [6] algorithm in case of normalized mutual information (top) and Modularity (bottom). 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.

7. 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 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. The future work will be to use adjacency list instead of matrix for genetic representation and the nature of our algorithm suggest that the proposed algorithm can be implemented in parallel system and both future work is expected to improve the time complexity.

Rererences

1. S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *ACM Transaction on Knowledge Discovery fromData*, 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 KnowledgeDiscovery and Data Mining, (KDD'07)*, pages 153–162, 2007.

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

5. M. Ehrgott. Multi-criteria Optimization. Berlin,Germany, 2005. Springer.

6. 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.

7. 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.

8. H. J., Nafpliotis, and G. DE. A niched pareto genetic algorithm for multiobjective optimization. In *Proc. of the first IEEE conference on evolutionary computation. IEEE world congress oncomputational intelligence*. Orlando, FL, USA, June 1994.

9. A. K. Jain, M. Murty, and P. Flynn. Data Clustering In *ACM Computing Surveys*, volume 31, pages 264–296, 1999.

10. S. JD. Multiple objective optimization with vector evaluated genetic algorithms. In *Proc. of theinternational conference on genetic algorithm and their applications*, 1985.

11. D. K, A. S, P. A, and M. T. A fast elitist non-dominated K. Deb, S. Agrawal and A. Pratap and T. Megarivan sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proc. of 6th international conference on parallel problemsolving on nature*, Paris,France, September 2000. Springer.

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 KnowledgeDiscovery 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 WideWeb Conference (WWW'08)*, pages 685–694, 2008.

16. S. N. and D. K. Multiobjective optimization using non-dominated sorting in genetic algorithms. 1994.

17. M. Newman and M. Girvan. Finding and evaluating community structure in networks. In *PhysicalReview E*, volume 69, pages 6113–6127, 2004.

18. H. P and lin C-y. Genetic search strategies in multi-criterion optimal design. pages 99–107, 1992.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

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