ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

Community Detection in a Social Network Using Differential Evolution with Multiple Objective Functions

Shubhankar Banerjee, Mrigank Shekhar, Nehal Agarwal
B.Tech Student, Dept of CSE, VIT University

Abstract*— Detection of community structure in a complex network has been increased considerably over the recent times. Many algorithms have been adduced so far, but none of them has been subjected to stringent tests to evaluate their performance. In this paper we have used differential evolution for detecting communities in complex networks. The advantage of using differential evolution over the other community detection algorithms is that it does not require any prior knowledge about the community structure, which is particularly useful for its application to real-world complex networks where prior knowledge is generally not available. After generation of a new population, the idea is to delete the 'n' unfit individuals, and to breed 'n' new ones from the fit individuals. Each population, therefore, needs to be awarded a figure of merit, to indicate how close it came to meeting the overall specification, and this is generated by applying the objective function to the population results. In this we have used five different types of objective function as a figure of merit and did a comparative study to reach a logical conclusion over the use of which real time methodology is most suitable for the desired result.

***Index Terms*—Clustering, Community Detection, Differential Evolution, Objective Functions, Social Network.**

I. INTRODUCTION

A social network is a social structure made up of a set of social actors (such as individuals or organizations) and a complex set of dyadic ties between these actors. The social network perspective provides a clear way of analyzing the structure of whole social entities. The study of these structures uses social network analysis to identify local and global patterns, locate influential entities and examine network dynamics. In general, social networks are self-organizing, emergent and complex, such that a globally coherent pattern appears from the local interaction of the elements that make up the system. These patterns become more appetent as network size increases. The modern science of networks has brought significant advances to our understanding of complex systems. The investigation of community structures has given rise to many diverse algorithm, most of them are unsuitable for large-scale real-time social networks because of the computational cost. Communities allow us to discover groups of interacting objects and the relations between them. Detecting communities is of great importance in sociology, biology and computer science, disciplines where systems are often represented as graphs. For example, in social networks, communities correspond to groups of friends who attended the same school, college, or who come from the same native place. In protein interaction networks, communities are functional modules of interacting proteins. The solution to this problem is not yet satisfactorily solved, despite the huge effort of a large interdisciplinary community of scientists working on it over the past few years. In this paper, we have used Differential evolution algorithm to detect network communities from raw network data. Differential evolution is a type of evolutionary algorithm which operate on a population of potential solutions applying the principal of the survival of the fittest to produce better and better approximation to a solution. Five different objective functions are used as modularity to find the optimal community structure of the network.

II. EXPLANATION OF THE METHODS

A. Differential Evolution

DE is an optimization technique which iteratively modifies a population of candidate solutions to make it converge to an optimum of the function. The initial population chosen randomly should cover the entire parameter space as much as possible. Here we have assumed a uniform probability distribution for all random decisions. DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector. This step is called Mutation. The mutated vector's parameters are then mixed with the parameters of another predetermined vector, the target vector known as the trial vector. This parameter mixing is known as "crossover". If the trial vector produces a lower cost function value than the target vector, then the trial vector is replaced by the target vector in the following generation. This operation is called selection.

B. Mutation Operation

After initialization, DE employs the mutation operation to produce a mutant vector Vi for each target vector xi,G

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

,where i =1,2,3,. .,NP. The mutant vector is generated by applying the following mutation strategy: vi, G +1= xr1,G + F (xr2,G - xr3,G). The randomly chosen integers r1, r2 and r3 are also chosen to be different from the running index, so that NP must be greater or equal to four to allow for this condition. F is a real and constant factor between [0, 2].These indices are randomly generated once for each mutant vector. The scaling factor is a positive control parameter for scaling the difference vector.

C. Crossover Operation

After the mutation phase, DE employs the binomial (uniform) crossover to each pair of the target vector $X_{i,G}$ and its corresponding mutant vector $V_{i,G}$ to generate a trial vector[11]:

$$u_{i,G}^{j} = \begin{cases} v_{i,G}^{j}, & \text{if } (\text{rand}_j[0,1) \le \text{CR}) \text{ or } (j = j_{\text{rand}}) \\ x_{i,G}^{j}, & \text{otherwise} \end{cases}$$
$$j = 1, 2, \ldots, D.$$

Here, rand $_{(j)}$ is the jth evaluation of a uniform random number. CR is the crossover probability and CR Ɛ[0,1]. CR is the crossover constant [0;1]which has to be determined by the user.

D. Selection Operation

If the values of some parameters of a newly generated trial vector exceed the corresponding upper and lower bounds, we randomly and uniformly reinitialize them within the prescribed range. Then, the objective function values of all trial vectors are evaluated. After that, a selection operation is performed. To decide whether or not it should become a member of generation G + 1, the trial vector $u_{i,G+1}$ is compared to the target vector $x_{i,G}$. If vector $u_{i,G+1}$ yields a smaller cost function value than $x_{i,G}$, then $x_{i,G+1}$ is set to $u_{i,G+1}$,otherwise.

III. DESCRIPTION OF OBJECTIVE FUNCTIONS EXPERIMENTED

A. Fitness Function

A fitness function is used to summarize, as a single figure of merit, how close a given design solution is to achieving the set aims. After each round of testing, or simulation, the idea is to delete the 'n' worst design solutions, and to breed 'n' new ones from the best design solutions. Each design solution, therefore, needs to be awarded a figure of merit, to indicate how close it came to meeting the overall specification, and this is generated by applying the fitness function to the test, or simulation, results obtained from that solution.

B. Clustering Coefficient Function

A clustering coefficient determines that which nodes in a graph will cluster together. Nodes having higher clustering coefficient tend to create tightly knit groups.

C. Betweenness Centrality

Betweenness centrality gives a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node. Betweenness centrality is a more useful measure of both the load and importance of a node. The Betweenness of a vertex v in a graph G:=(V,E) with V vertices is computed as follows:

1. For each pair of vertices (s,t), compute the shortest paths between them.
2. For each pair of vertices (s,t), determine the fraction of shortest paths that pass through the vertex in question (here, vertex v).
3. Sum this fraction over all pairs of vertices (s,t).

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

D. Closeness Centrality

In connected graphs the length of the shortest path between all pairs of nodes is known as their natural distance metric. The farness of a node is defined as the sum of its distances to all other nodes, and the inverse of the farness is defined as its closeness. Thus, the more central a node is the lower its total distance to all other nodes. Closeness can be regarded as a measure of how long it will take to spread information from *s* to all other nodes sequentially.

E. Degree Centrality

Degree centrality is the simplest centrality measure, which can be defined as the number of links incident upon a node (i.e., the number of ties that a node has). The degree centrality of a vertex v , for a given graph G:= (V ,E) with $|V|$ vertices and $|E|$ edges, is defined as : $C_D(v) = \deg(v)$.

IV. DESCRIPTION OF DATASET UNDER STUDY

A. ZACHARCY – KARATE CLUB

These are data collected by members of a university karate club by Wayne Zachary. The ZACHE matrix represents the presence or absence of ties among the members of the club the ZACHE matrix indicates the relative strength of the associations (number of situations in and outside the club in which interactions occurred).

B. NBLOGGERS

It is a 32x32 Matrix of bloggers.

C. STRAYER, CUMINS – DOMINANCE AMONG ADULT MACAQUES

A 30x30 matrix of dominance encounters. This matrix is built from experimental data on the 30 adult members (15 males & 15 females) of a community of macaque monkeys (Macacamulatta) housed at the Wisconsin Regional Primate Center in Madison. The monkeys were deprived of water and

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

then observations of which monkey got access to a drinking fountain were made.

D. SUNDARESAN, FISCHOFF, DUSHOFF, RUBENSTEIN –ZEBRA AFFILIATION

This matrix is based on a study of a community of 28 Grevy's zebras. Cell entries of 1 indicate that a pair of zebras appeared together at least once during study. Entries on 2 indicate a statistically significant tendency of pairs to appear together.

V. IMPLEMENTATION AND RESULT

Code has been implemented in C++ language. One can select the dataset and the type of objective function he/she wants to use in the program from a menu driven screen as input.

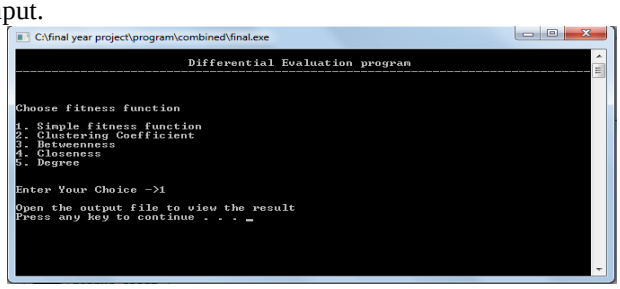

Fig. 1 Input Screen

Output is in the form of a file which shows the number of clusters formed and the individual number and its genes.

```
        Differential evolution using simple fitness function
-------------------------------------------------------------------------------
Individual ->0 Genes ->14 11 17 8 10 7 13  Fitness value ->747

Individual ->26 Genes ->14 8 13 7 17 11 1  Fitness value ->735

Individual ->27 Genes ->14 11 17 7 10 1 13  Fitness value ->734

Individual ->640 Genes ->1 8 14 17 10 11 13  Fitness value ->720

Individual ->658 Genes ->10 13 17 14 8 7 1  Fitness value ->713

Individual ->1308 Genes ->14 10 11 1 8 7 13  Fitness value ->709

Individual ->1399 Genes ->14 8 7 4 17 11 13  Fitness value ->704

Individual ->1548 Genes ->11 8 13 1 10 17 7  Fitness value ->701

Individual ->1803 Genes ->14 11 17 10 1 7 8  Fitness value ->695

Individual ->1819 Genes ->14 10 11 4 17 7 13  Fitness value ->691

Individual ->2122 Genes ->14 11 17 13 10 7 5  Fitness value ->683

Individual ->2168 Genes ->14 11 17 8 10 7 4  Fitness value ->676

Individual ->2173 Genes ->14 10 6 7 17 11 13  Fitness value ->675
Individual ->2177 Genes ->17 14 7 4 11 13 1  Fitness value ->672

Individual ->2557 Genes ->14 8 1 4 17 11 13  Fitness value ->671

Individual ->2558 Genes ->14 7 17 5 1 11 13  Fitness value ->669
```

Fig 2 Output File

Code has been complied on Windows environment using DevC++ editor. The program was executed with a population size of 10,000 individuals. Figure 2 shows the result obtained by executing the program using the NBLOGGERS dataset and fitness function as objective function. Only those generations are displayed that showed improvement in fitness. Duplicate fitness has been removed. Table 1.shows a comparative results obtained by executing the program on different datasets and applying five different objective functions to each of the datasets.

Dataset	Fitness function	Clustering Coeff	Betweenness	Closeness	Degree
NBloggers	Max. Value = 747 Min Value = 643 No. of Cluster = 33	Max. Value = 1.976 Min Value = 1.778 No. of Cluster = 10	Max. Value = 89.5 Min Value = 77 No. of Cluster = 14	Value = 0.235 No. of Cluster = 1	Max. Value = 9.71e+009 Min Value = 9.34e+009 No. of Cluster = 2
Zachary	Max. Value = 61 Min Value = 58 No. of Cluster = 2	Value = 0.161 No. of Cluster = 1	Max. Value = 767 Min Value = 719 No. of Cluster = 10	Value = 0.2081 No. of Cluster = 1	Value = 1.105e+010 No. of Cluster = 1
Zebra	Value = 41 No. of Cluster = 1	Value = 0.108 No. of Cluster = 1	Max. Value = 204.7 Min Value = 192.4 No. of Cluster = 10	Value = 0.207 No. of Cluster = 1	Max. Value = 9.6622e+009 Min Value = 9.49248e+009 No. of Cluster = 24
Dominance	Max. Value = 119 Min Value = 112 No. of Cluster = 8	Value = 0.309 No. of Cluster = 1	Value = 0 No. of Cluster = 1	Value = 0.215 No. of Cluster = 1	Max. Value = 8.03271e+009 Min Value = 7.62961e+009 No. of Cluster = 75

Table 1

From the above table, it can be observed that for the NBloggers dataset Fitness function gives 33 clusters, whereas clustering coefficient and Betweenness gives 10 and 14 clusters respectively. Both The algorithm here converges in same manner. Whereas Convergence is very high for objective functions closeness and degree which gives 1 and 2 clusters respect. So here we can infer that more the difference between the maximum and minimum value of a function, more is the number of clusters formed and less is the accuracy. Similarly the table shows the results for other datasets also.

VI. CONCLUSION AND FUTURE SCOPE

In this paper the efficiency of differential evolution for solving community detection problem has been investigated. It is evident from the study that the efficiency of differential algorithm depends upon the type of Objective function used. The paper enlists the advantages and implementation techniques of differential algorithm and their usefulness in solving community detection problem. It also mentions the extent and approach of evolutionary algorithms and the role of objective functions in it. The experimental results have demonstrated that DE is very effective in community detection in complex networks, including those with very vague community structures. In addition to its excellent performance, another merit of DE is that it does not require any prior knowledge about the community structure when detecting communities in a complex networks. The limitation of this work is that we have only used modularity as the objective function to find the optimal community structure of a network. However it has been recently pointed out that such approach might suffer from a so called resolution limit problem i.e some modules smaller than a specific scale will not be detected by the algorithms that only optimize modularity. We will further investigate this problem in future work.

REFERENCES

[1] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, Proceedings of National Academy of Science of the United States of America 99 (2002) 7821 – 7826.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

[2] M. Tasgin, H. Bingol, Community detection in complex networks using genetic algorithm, arXiv: 0711.0491, 2007.

[3] S. Fortunato, Community detection in graphs, Physics Reports 486 (2010) 75 – 174.

[4] F. Wu, B. A. Huberman, Finding communities in linear time: A physics approach, European Physical Journal B 38 (2004) 331 – 338.

[5] G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structures of complex networks in nature and society, Nature 435 (2005) 814 – 818.

[6] Newman, M. E. J. and M. Girvan. 2004. Finding and evaluating community structure in networks. Physical Review E 69: 026113.

[7] Radicchi, F., C. Castellano, C., F. Cecconi, V. Loreto, and D. Parisi. 2004. Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the United States of America 101: 2658-2663.

[8] Bader, G.D., Hogue, C.W. An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics, 4:2, 2011.

[9] Barabasi, A.L., Oltvai, Z.N. Network biology: understanding the cell's functional organization. Nature Reviews, 5:101–113, 2009.

[10] Beyer, K., Goldstein, J., Ramakrishna, R., Shaft, U. When is "nearest neighbor" meaningful? In Proceedings of 7th Int. Conf. on Database Theory (ICDT), 2007.

[11] Guanbo Jia, Zixing Cai, Mirco Musolesi, Yong Wang, Dan A. Tennant, Ralf. J.M. Weber, John K. Heath2, and Shan He. Community Detection in Social and Biological Networks using Differential Evolution, 2012.

AUTHOR'S PROFILE

Mr. Shubhankar Banerjee is currently pursuing his BTech in Computer Science and Engineering from Vellore Institute of Technology, Vellore. He is a passionate developer and just loves to code. He is also a freelance web developer.

Mr. Mrigank Shekhar is UG Engineering student in Final Year BE (Computer Science and Engineering) from VIT University Vellore. His research interests are in Artificial Intelligence, Algorithm Design, Knowledge Discovery in Data and Web development.

Mr. Nehal Agarwal is UG Engineering student in Final Year BE (Computer Science and Engineering).He is doing is Final Year Project on Community detection in VIT University Vellore.