Proc. of The Sixth Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2017

ISBN: 978-1-63248-117-7 doi: 10.15224/ 978-1-63248-117-7-51

COMMUNITY DETECTION USING Central Force Optimization (CFO)

Govind Agarwal[1], Santosh Kumar Chourasia[1], Anupam Biswas[1], Siddhartha K Arjaria[2], Bhaskar Biswas[1]

***Abstract*— Community structure is believed to be one of the notable features of complex networks representing real complicated systems. Very often, uncovering community structures in networks can be regarded as an optimization problem, thus, many evolutionary algorithms based approaches have been put forward. . In this paper Central Force Optimization (CFO), physics based optimization, and its variants are used to detect communities. CFO is deterministic in nature, unlike the most widely used meta-heuristics. However, CFO is not free from the problem of premature convergence. Therefore the variants used Adaptive CFO (ACFO) and Multi-Start CFO (MCFO) are used in enhancing the convergence.**

Keywords—*Community Detection, Physics Inspired Optimization, CF, Social Network Analysis.*

I. Introduction

In recent times, Social networks became an important role in every aspect of modern life. Researchers, scientists and industry have shown interest in analysis of such networks. These Networks can be viewed as massive graphs that are composed of a set of vertices and edges, where nodes represent the objects and links represent the interactions amongst them, and then apply certain techniques to analyze the properties of the graph based network. One of such techniques for analyzing social networks is community detection. Communities are regarded as sub-graphs which have dense intra-links and sparse inter-links. Communities are also known as clusters or modules of graph. Mostly, the detection of community structures in networks can be considered either as a clustering problem or an optimization problem, thus, the choice of an appropriate optimization algorithm and evaluation function affects the ultima detection performance.

Over the last decades, many nature-inspired heuristic optimization algorithms without requiring much information about the function became the most widely used optimization methods such as genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO), cuckoo search (CS) algorithm, group search optimizer (GSO), and glowworm swarm optimization (GSO1).

1. Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi India.

2. Technocrats Institute of Technology, Bhopal, India

These search methods all simulate biological phenomena. For the algorithms that are not bio-inspired, most have been developed by mimicking certain physical and/or chemical laws, including electrical charges, gravity, river systems, etc. As different natural systems are relevant to this category, we can even subdivide these into many subcategories which are not necessary. Since our work is based on physics inspired optimization techniques, we will not talk about chemistry or biology inspired algorithms.

Some of the well-known physics optimization algorithms are as follows [5]:

1. Newton's gravitational law
   - ➢ Pure physics
     - CFO
     - APO
     - GSA
     - GIO
   - ➢ Semi physics
     - IGOA
2. Quantum mechanics
   - ➢ Pure physics
     - QGA
     - QEA
   - ➢ Semi physics
     - QPSO
     - QSE
     - QICA
     - CQACO
     - QBSO
3. Universe theory
   - ➢ Pure physics
     - BB-BC
     - GBSA
4. Electromagnetism
   - ➢ Pure physics
     - EM
5. Electrostatics
   - ➢ Pure physics
     - ES

Different from these algorithms, some heuristic optimization algorithms based on physical principles have been developed, for example, simulating annealing (SA) algorithm, electromagnetism-like mechanism (EM) algorithm, central force optimization (CFO) algorithm [1], gravitational search algorithm (GSA), and charged system search (CSS). SA simulates solid material in the annealing process. EM is based on Coulomb's force law associated with electrical charge process. GSA and CFO utilize Newtonian mechanics law. CSS is based on Coulomb's force and Newtonian mechanics laws. Unlike other algorithms, CFO is a deterministic method. In other words,

Proc. of The Sixth Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2017

ISBN: 978-1-63248-117-7 doi: 10.15224/ 978-1-63248-117-7-51

there is not any random nature in CFO.

As described in [1] [2], Central Force Optimization (CFO) is a new deterministic multi-dimensional search metaheuristic based on the metaphor of gravitational kinematics. It models "probes" that "fly" through the decision space by analogy to masses moving under the influence of gravity. Equations are developed for the probes' positions and accelerations using the analogy of particle motion in a gravitational field. In the physical universe, objects traveling through three dimensional spaces become trapped in close orbits around highly gravitating masses, which is analogous to locating the maximum value of an objective function. In the CFO metaphor, "mass" is a user defined function of the value of the objective function to be maximized. The CFO may get easily trapped in a local optimum when solving complex problem. To address shortcomings of traditional CFO, two variants of CFO, Multi-start CFO Adaptive CFO, has been proposed and has been used in this work. The other variants of CFO suffer from following two shortcomings. Firstly, many exciting CFO variants avoid premature convergence at the cost of high computational burden. Secondly, some modifications of CFO are restricted to low-dimensional problems.

II. Proposed Method

In this section, by taking the advantage of network prior knowledge, we have algorithms for community detection using simple CFO, Adaptive CFO and Multi-Start CFO. The proposed algorithm is evaluated on various metrics, which will be discussed later. All abbreviations, variables and equations are adopted as in [1], [2] and [5]. This section will illustrate the advanced algorithm in detail.

2.1. Community Detection Using Simple CFO

Algorithm 1: Community detection using simple CFO

Parameters: Nt (Number of iterations), Np (Population Size), G (Gravitation Constant), α, β.

Input: Network Adjacency Matrix

Output: Best fitness, Partition of the network into the communities.

1. Begin
2. Population initialization: initialize position vectors X, i.e.,
3. pop.X ← label_propagation; initialize acceleration vectors A
4. pop.A = 0; calculate particles' fitness pop.fitness;
5. Update current best position vectors pbest and global best position vectors gbest;
6. **For** iter = 0; iter < N_t; iter + +
7. **For** each particle i ∈ N_p **do**
8. update particle i's status according to [1] [2];
9. rearrange particle i's position vector;
10. evaluate particle i's fitness;
11. update particle i's pbest vector;
12. **End**
13. update the gbest vector;
14. **End**
15. **End**

The population initialization step needs to initialize particle's' position and velocity vectors and update the pbest and gbest vectors. To initialize the position vectors, a novel mechanism based on label propagation is introduced. This method can generate individuals with higher clustering efficiency [3]. The pbest vectors are initialized the same as the position vectors and the gbest vector is set as the best position vector in the current population.

In this paper, we redefine the term ''position'' and ''velocity'' under discrete context. The definitions are as follows:

Definition of discrete position: the position vector represents a partition of a network. We define a position permutation in the form of $X = \{x_1, x_2, \ldots, x_n\}$ where $x_i \in [1, n]$ is an integer and n is the number of the nodes of a network. If $x_i = x_j$, then we take it that node j and i are in the same community. A graphical illustration of the particle representation schema is shown in Fig. 1. The above position definition does not need to know about the topology structures of the networks, which makes it transplantable to other kinds of networks. The representation schema is easy to decode which will lower down the computational complexity and it can automatically determine the communities of the network, what is more, it is easy to implement label propagation operation to generate initial solutions with high clustering precision.

Definition of discrete acceleration: we define the acceleration permutation of a particle as $A = \{a_1, a_2, \ldots, a_n\}$ where $a_i \in \{0, 1\}$. If $a_i = 1$, then the corresponding element x_i in the position vector X will be changed according to Eq. (1), otherwise, x_i keeps unchanged.

$$X^p(t+1) \leftarrow X^p(t) \ominus A^p(t) \quad (1)$$

Definition of particle status update rules

In Simple CFO, a particle adjusts its acceleration by the force exerted by its neighbours. The new acceleration helps the particle to fly to promising region. The particle status update rules defined in the canonical version of CFO no longer fit the discrete context, therefore, based on the newly defined particle status, a redefinition of update has been done.

Particle position rearrangement:

In our proposed algorithm, a particle' position has been defined as an integer permutation. Each position permutation corresponds to a community structure of a network. However, there exists the situation that different position permutations are structurally equivalent, i.e., they correspond to the same community structure. For example, given two position vectors P1 = {3 1 3 1 3 1} and P2 = {6 4 6 4 6 4}, then according to our defined particle representation schema, P1 and P2 are structurally equivalent

Proc. of The Sixth Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2017

ISBN: 978-1-63248-117-7 doi: 10.15224/ 978-1-63248-117-7-51

because they both correspond to the same network partition that nodes 1, 3 and 5 belong to one community and nodes 2, 4 and 6 belong to another community [5].

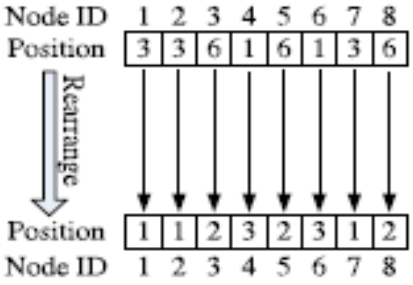

Figure 1: Rearrangement Operation

To avoid unnecessary computing, we design a particle position rearrangement operation in our algorithm. A graphical illustration of the rearrangement operation is shown in Fig. 1. In Fig. 1, when we decode the original position vector we will divide the eight nodes into three communities, i.e., c1 = {1, 2, 7}, c2 = {3, 5, 8} and c3 = {4, 6}. Then we rearrange the position in the following way: the 1st, 2nd and 7th elements in the vector have been changed to be 1; the 3rd, 5th and 8th elements have been changed to be 2; the 4th and the 6th elements have been changed to be 3.

2.2. Community Detection Using Adaptive CFO

The algorithm for Community Detection Using Adaptive CFO is same as Community Detection Using Simple CFO except for the updating part.

Algorithm 2: Community detection using adaptive CFO

Parameters: N_t (Number of iterations), N_p (Population Size), G (Gravitation Constant), α, β.
Input: Network Adjacency Matrix
Output: Best fitness, Partition of the network into the communities.

1. Begin
2. Population initialization: initialize position vectors X, i.e.pop.X ← label_propagation;
3. initialize acceleration vectors A, i.e., pop.A = 0;
4. calculate particles' fitness pop.fitness;
5. initialize vectors V, i.e.. pop.V = 0;
6. update current best position vectors pbest and global best position vectors gbest;
7. **For** iter = 0; iter < N_t; iter + +
8. **For** each particle i ∈ N_p do
9. update particle i's status according to Equations in [2];
10. rearrange particle i's position vector;
11. evaluate particle i's fitness;
12. update particle i's pbest vector;
13. **End**
14. update the gbest vector;
15. **End**
16. **End**

Except for the update part, rest all the steps are same as Community Detection using Simple CFO.

2.3. Community Detection Using Multi- Start CFO.

Algorithm is also very similar to Simple CFO except that it is run twice using different initialization technique in each case. Then the best results of both iterations are compared and the better of the two is given as output with the corresponding partition of the network. Algorithms for the two initialization techniques are given below:

Algorithm 3: Initialization 1

Parameters : N_p (Population Size), N_d (Graph Size)

1. Begin
2. First Initialize a $N_d \times N_p$ matrix with first row =1, second row =2,..;
3. **For** each column vector
4. Choose a random value from 1 to N_d and assign it to x
5. **For** 1 to x
6. Choose a random value from 1 to N_d and assign it to y;
7. Change the community label of any adjacent vertex of y to the value same as y
8. **End**
9. **End**

Algorithm 4: Initialization 2

Proc. of The Sixth Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2017

ISBN: 978-1-63248-117-7 doi: 10.15224/ 978-1-63248-117-7-51

Parameters : N_p(Population Size), N_d(Graph Size)

1. Begin
2. Initialize a $N_d \times N_p$ matrix with all zeroes
3. **For** each column vector
4. **While** all nodes are labelled
5. Choose any non-labelled node
6. Do a BFS of length 2 from this node and assign the same label to the nodes traversed.
7. **End**

8. End

III. Experimental Results

The algorithms described are implemented and results obtained are evaluated on two metrics, Modularity and Coverage, which are widely used. Modularity is one measure of the structure of networks or graphs. It was designed to measure the strength of division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. Modularity is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random. The total coverage of the clusters is the total number of nodes covered by these clusters.

The Data Set used in experiments are shown below:

1. Strike Matrix which has 24 nodes
2. Karate Matrix which has 34 nodes
3. Dolphin Matrix which has 62 nodes
4. Football Matrix which has 115 nodes

These data sets are standard data sets which are used in evaluation of every community detection technique. The parameters values that are used during the experiments is listed in table 1.

Table 1: Parameter values for all the methods

Parameter	Value
G_{max}	10
N_t	100
N_p	N_d
A	1
B	0.5

Based on the above parameters, algorithms described are implemented on the data set described above. The results so obtained are shown below.

Table 2: Result for the modularity metric

Data Set \Algorithm	Simple CFO	Adaptive CFO	Multi- Start CFO
Strike Matrix	0.8172	0.8172	0.8172
Karate Matrix	0.6153	0.6153	0.7009
Dolphin Matrix	0.5785	0.6079	0.8007
Football Matrix	0.6310	0.7079	0.8059

Table 3: Result for the coverage metric

Data set\Algorithms	Simple CFO	Adaptive CFO	Multi- Start CFO
Strike Matrix	1	1	1
Karate Matrix	0.8846	0.8846	1
Dolphin Matrix	0.6226	0.6918	0.9371
Football Matrix	0.8613	0.7684	0.8891

IV. Conclusion and Future Work

By the results obtained above Community Detection Using Multi- Start CFO works better than Adaptive CFO. Community Detection using Adaptive CFO works better than Simple CFO. And due the deterministic nature of CFO, we can be sure that the results will remain, if the methods are executed repeatedly. Further, we have absorbed that the community detection using multi- start CFO works better especially for larger graphs in all the metrics used for evaluation of communities. The proposed methods can also be evaluated on larger data set or other metrics. It can also be compared with other biology based or physics based discrete optimization techniques.

References

[1] Liu, Y., & Tian, P. (2015). A multi-start central force optimization for global optimization. *Applied Soft Computing*, *27*, 92-98.

[2] Formato, R. A. (2007). Central force optimization: a new metaheuristic with applications in applied electromagnetics. *Progress In Electromagnetics Research*, *77*, 425-491.

[3] Cai, Q., Gong, M., Shen, B., Ma, L., & Jiao, L. (2014). Discrete particle swarm optimization for identifying community structures in signed social networks. *Neural Networks*, *58*, 4-13.

[4] Fortunato, S. (2010). Community detection in graphs. *Physics reports*,*486* (3), 75-174.

[5] Biswas, A., Mishra, K. K., Tiwari, S., & Misra, A. K. (2013). Physics-inspired optimization algorithms: a survey. *Journal of Optimization*, *2013*.

[6] Wang, M., Wang, C., Yu, J. X., & Zhang, J. (2015). Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework. *Proceedings of the VLDB Endowment*, *8*(10), 998-1009.

[7] Fister Jr, I., Yang, X. S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*.

[8] Chen, M., Kuzmin, K., & Szymanski, B. K. (2014). Community detection via maximization of modularity and its variants. *Computational Social Systems, IEEE Transactions on*, *1*(1), 46-65.

[9] Porter, M. A., Onnela, J. P., & Mucha, P. J. (2009). Communities in networks. *Notices of the AMS*, *56*(9), 1082-1097.

[10] Qian, W., Wang, B., & Feng, Z. (2015). Adaptive Central Force Optimization.*Mathematical Problems in Engineering*, *501*, 914789

About Author (s):

Dr.Siddhartha Kumar Arjaria is currently working as professor at Technocrat Institute of Technology, Bhopal. He obtained his Ph.D from MANIT,Bhopal. His field of interest is opinion mining, soft computing and AI techniques.

Proc. of The Sixth Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2017

ISBN: 978-1-63248-117-7 doi: 10.15224/ 978-1-63248-117-7-51

Govind Agarwal graduated from Indian Institute of Technology (BHU) in 2016. Currently he is working at Amazon.

Santosh Kumar Chourasia[1] graduated from Indian Institute of Technology (BHU) in 2016. Currently he is working at Paytm.

Anupam Biswas received M.Tech. degree in Computer Science and Engineering from Motilal Nehru National Institute of Technology Allahabad.

He is working toward the Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi. His research interests include Fuzzy Systems, Data Mining, Link Prediction, Social Network Analysis, Evolutionary Computation and Optimization.

Bhaskar Biswas received Ph.D. in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi He is working as Assistant Professor at Indian Institute of Technology (BHU),Varanasi in the Computer Science and Engineering department. His research interests include Data Mining, Text Analysis, Machine Learning, Link Prediction, Social Network Analysis.