International Journal of Future Generation Communication and Networking
Vol. 11, No. 2 (2018), pp.33-48
http://dx.doi.org/10.14257/ijfgcn.2018.11.2.04

Finding Communities in Social Networks with Node Attribute and Graph Structure using Jaya Optimization Algorithm

Mini Singh Ahuja[1] and Jatinder Singh[2]

[1]Research Scholar, Punjab Technical University, Jalandhar
[2]Professor, Baba Bhag Singh University, Jalandhar
minianhadh@yahoo.co.in, bal_jatinder@rediffmail.com

Abstract

The complex systems can be denoted as networks where subsets of nodes are more connected among themselves. Here, the major problem is the detection of community structure in complex networks like social networks, web networks, telecommunication networks, biological, and trade networks. Community detection allows computing groups of interacting objects and the relations between them. Almost all the researches done in this area have taken into account the graph structure of the network for finding the communities. In this paper, we propose a multi objective community detection (MCD) method for complex networks using new optimization algorithm which will consider graph structure and the content associated with the nodes. Multiple metrics are used for detecting communities in these networks. These are modularity, generalized modularity and similarity of attributes. The time varying metrics are optimized by the proposed evolutionary combined Jaya optimization algorithm, which achieves exact detection without affecting execution time. The performance of proposed method has been analyzed with the five different networks. The experimental result shows that the proposed method achieves favorable results which are quite superior to existing methods.

Keywords: *complex network, community detection, evolutionary combined Jaya algorithm, execution time*

1. Introduction

Many complex systems in real world can be represented as complex networks such as social networks, web networks, telecommunication networks, biological networks, technology and trade networks etc. The property of community structure refers to the characteristics of dense intra-community connections and sparse intercommunity connections which largely exist in complex networks. The content of community structure varies according to different application fields. For example, it refers to people sharing similar characteristics in social networks and the bio-modules with similar functions in biological networks. The purpose of community detection in complex networks is to reveal community structure inherent in those complex networks. The great significance both theoretically and practically, has attracted many researchers of various disciplines. The current work in complex networks science is based on static networks, which are graphs where vertices and edge are fixed permanently. However, the diverse interactions that make up empirical complex networks are often quite fluid where new connections are formed often, while others dissolve, providing opportunities for topological changes that can have a major impact on dynamics of the network.

Girvan and Newman (GN) algorithm marked the beginning of the era and considered the property of community structure, in which network nodes are joined together in tightly knit groups, between which there are only loose connections. The spectral

Received (January 5, 2018), Review Result (March 2, 2018), Accepted (March 7, 2018)

ISSN: 2233-7857 IJFGCN

clustering based on graph partitioning algorithm, has been implemented using a hand full of software commands. A realistic benchmark test for community detection algorithm for complex networks considered the effect of heterogeneity of community size in observed real networks. The Newman's fast community detection algorithm tends to favor the creation of large communities at the expense of smaller ones. The fast algorithm ensures that communities of differing sizes are treated on an equal footing, thus side-stepping this problem. A mathematical programming approach also named as mixed integer quadratic programming (MIQP) [1] has been used to identify the optimal community structures by maximization of network modularity metric for partitioning a network into modules. ComTector method [2] for community detection is based on the overlapping nature of cliques in our real world; this algorithm can be applied to many large sparse graphs. It is very simple and intuitive to extract satisfactory results on networks whose community structures are known before. The q-state Potts community detection method [3] handles a parameter as threshold tuning and selects proper value using a priori principle. A guided community detection algorithm [4] is based on the class structures of object-oriented software to obtain the optimized community structures in the method-method networks, which also corresponds to optimized class structures. The community detection using representative energy method [5] is used to compute community with responsible energy consumption terms. An overlapping community detection algorithm [6] computes the seed sets by the spectral partition and then extends them with a special random walks technique. A nonparametric Bayesian generative model [7] is used to compute the intuitive of communities using Markov chain Monte Carlo (MCMC). A latent Dirichlet allocation (LDA) based link partition (LBLP) method [8] has been used to compute the communities with an adjustable range of overlapping. A clustering based community detection approach uses the detection metric as modularity and optimizes it by reducing to a minimum cut problem and graph partitioning. An unbounded, and user configurable parallel k-clique community detection method [9] uses the innovative technique to compute the local components of a network. Dynamic multi objective genetic algorithm [10] discovers the communities in dynamic networks with temporal smoothness. A local edge anti-triangle centrality (EACH) approach [11] utilizes the centrality for community detection. EACH was characterized by being free of any parameters including the prior number of the expected communities and independent of any additional measures to decide the community structure. The virtue of density based algorithms and rough set models are used to design a density-based rough set model (DBRSM).

Contributions. In this paper, we propose a multi objective community detection method for social networks. The proposed method considers the time varying constraints such as modularity, generalized modularity and similarity of attributes. The time varying constraints are optimized by the proposed evolutionary combined Jaya optimization algorithm, which minimize the execution time and repeated solution.

The remaining paper is organized as follows. The literature review related to our contribution is present in Section 2 the Jaya Algorithm is present in Section 3. The Section 4 describes the problem methodology and model of proposed method. In Section 5 discuss the detailed work of proposed method and the experimental results are given in Section 6. Finally, the paper concludes in Section 7.

2. Literature Review

Hao *et al.*, [12] has proposed a k-clique community detection algorithm using formal concept analysis (FCA). A formal context was constructed from a given social network by a modified adjacency matrix and a type of special concept named k-clique concept, uses the same k-size of extent and intent in a formal concept lattice. Then, the k-clique detection problem became equivalent to the k-clique problem. The algorithm has achieved a higher F-measure value rather than existing algorithms.

Ma *et al.*, [13] has proposed evolutionary nonnegative matrix factorization (ENMF) framework for detecting dynamic communities. The semi-supervised ENMF (sE-NMF) function combines the ENMF and spectral clustering based on priori information. sE-NMF considers a general framework for the semi-supervised evolutionary clustering. The algorithm escapes the local optimal solution without increasing time complexity. Experiments had been conducted on number of artificial and real world dynamic networks to show that this method is not only more accurate but also more robust than the state-of-the-art approaches.

Wang *et al.*, [14] has introduced a two-stage approach for the important cyber-security problem of detecting the compromised nodes. A flow-based and graph based approach has been used to detect anomalies using resultant leveraging large deviations of an empirical distribution. The quantization flow has been computed from the flow based approach and computation of the distribution of node interactions by the graph based approach. Then, this information detects the bots using ideas from social network community detection in a graph that captures correlations of interactions among nodes over time.

Huang *et al.*, [15] has proposed meta-heuristic algorithm which combines the line graph, ensemble learning, and particle swarm optimization (LEPSO) for community detection in social networks. A discrete PSO consists of an encoding scheme with ordered neighbors and a particle updating strategy with ensemble clusters for improving the optimization ability to search communities hidden in social networks. The post processing strategy has been used to merge the finer-grained and suboptimal overlapping communities in the complex social networks.

Mahmood *et al.*, [16] has proposed a geodesic density gradient (GDG) algorithm using the information gathered from the network nodes. Position of different nodes in the geodesic space encodes the network structure to compute the density at a specific point in all fields induced by all nodes. The direction of positive density gradient uses an iterative algorithm till each node reaches a local maximum. The nodes that drift to the same region of space belong to the same communities in the original network. The direction of movement and final position of each node is used as an important clue for community membership assignment.

Alik *et al.*, [17] has proposed a rule based community detection (RUCA) for label propagation using static information. The RUCA algorithm when applied on multiple networks detects the number of communities, discovers isolated nodes, isolated communities, and always outputs stable partitions. The RUCA algorithm uses the network structure that guides the progress and does not optimize any specified function or measure of community strength. The rules define: how each node decides to which community of its neighboring nodes to join. The rules consider the number of neighbors that belong to the same community and the similarity of each node with its neighbors. The rules have enhanced the label propagation that achieves higher speed of execution and improves the quality of the communities detected.

Interdonato *et al.*, [18] has proposed local community detection in multilayer networks (ML-LCD) realized by the internal and external connectivity. The function optimized by a multilayer local community has been instantiated into three methods that differ in the way. The extensive analysis over seven real-world multilayer networks provides the significance and ability to detect multilayer local communities. Local communities are detected by a non-overlapping community structure in the complex network. The ML-LCD method computes the higher-modularity communities than state-of-the-art methods designed for multilayer global community detection.

Tagarelli *et al.*, [19] has proposed a modularity-driven ensemble-based approach to multilayer community detection. A key aspect is that it computes consensus community structures that not only capture prototypical community memberships of nodes, but also preserves the multilayer topology information and optimizes the edge connectivity in consensus. Empirical evidence obtained on seven real-world multilayer networks sheds light on the effectiveness and efficiency of modularity-driven ensemble-based approach. This method outperforms state-of-the-art multilayer in terms of modularity and redundancy assessment criteria.

Li *et al.*, [20] has proposed a regularized semi-nonnegative matrix tri-factorization (ReS-NMF) community detection algorithm. The ReS-NMF algorithm captures communities by mapping the signed network into a latent space and improves the detection accuracy by studying the effects of negative links and enforcing the sparsity constraints on the indicator matrix. The effects of the negative links are used to detect the communities and have introduced a graph regularization to distribute the pair of nodes which are connected by negative links into different communities.

3. Jaya Algorithm

Jaya is a simple and powerful optimization algorithm proposed by R Rao in 2015 for solving constrained and unconstrained optimization problems. Jaya has been retrieved from the Sanskrit word ‘Jaya’ which means Victory. The Jaya algorithm always moves towards the global best solution rather than a local best solution in a large dataset. Jaya algorithm does not require any algorithm specific control parameter where as other evolutionary algorithms require common controlling parameters like population size, number of generations etc along with proper tuning of algorithm specific parameters. Moreover, it requires only one phase as it uses only one single equation which makes it simpler for solving constrained and unconstrained optimization problems.

Let *f(x)* is the objective function which is to be maximized (or minimized). At any iteration i, let there be m number of design variables (j= 1,2…m), n number of candidate solutions, best be the best value with maximum fitness and the worst be the worst value with minimum fitness in the entire candidate solutions. If $X'_{j,k,i}$ is the value of the jth variable for the kth candidate during ith iteration, then value is updated using the equation

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\ (X_{j,best,i} - |X_{j,k,i}|) - r_{2,j,i}\ (X_{j,worst,i} - |X_{j,k,i}|)$$

Where $X_{j,best,i}$ is the value of the variable j for the best candidate and $X_{j,worst,i}$ is the worst value of the variable j for the worst candidate. $X'_{j,k,i}$ is the updated value of $X_{j,k,i}$ and $r_{1,j,i}$ and $r_{2,j,i}$ are the two random numbers for the jth variable during the ith iteration in the range [0,1]. The term $r_{1,j,i}\ (X_{j,best,i} - |X_{j,k,i}|)$ tells that solution is moving closer to best solution and term $r_{2,j,i}\ (X_{j,worst,i} - |X_{j,k,i}|)$ tells that solution is moving far from the worst solution. $X'_{j,k,i}$ (Updated value) is accepted only if it’s better function value. At the end of each iteration, all accepted function values are maintained and later these values become input to next iteration. The Jaya algorithm is based on the idea that it always moves towards the best solution and moves away from the worst solution.

4. Problem Methodology and Proposed Model

4.1. Problem Methodology

Reihanian *et al.*, [21] has proposed a multi-objective discrete biogeography based optimization (BBO) algorithm to compute the communities. This algorithm uses the Pareto-based approach for community detection using a new metric, SimAtt along with Modularity for detecting communities [22]. As the two objective functions have to be maximized, a two phase sorting strategy has been introduced for non-dominated sorting and crowding-distance to sort the generated solution of a population. For mutation probability approximation, it uses a chaotic mechanism to dynamically tune the mutation probability in each iteration. A set of non-dominated solutions have been used to compute the best compromise solution among these non-dominated ones.

The problem of network representation is very complex and contains different variants in the traditional methods. Many works [12]-[21] have used community detection metric as modularity and its variants. These focus only on the graph structure of networks for finding network communities. But nodes have associated attributes as well. For example, Facebook includes photos, comments and other shared objects. So finding communities only on the basis of modularity will not be fruitful. Moreover, modularity maximization is suitable only for uniparite networks. Modularity is not suitable for high density networks because the degrees of vertices are not equal for all cases and they do not share any edges together. A high density network consists of vertices with different relations *i.e.*, multi-dimensional and so they need multiple input constraints for community detection. Generalized modularity can take advantage both of vertex similarity and path length.

For this reason, we propose multi objective community detection (MCD) method for social networks based on the evolutionary combined Jaya optimization algorithm. The proposed method will consider both graph structure and the content associated with the nodes in the form of attribute of similarity for detecting communities. In proposed method, we consider multiple constraints. These are modularity, generalized modularity, and similarity of attributes for community detection. Generalized modularity takes into account common neighbors and longer paths between vertices and compares the number of these paths to a randomly distributed network, which achieves a more comprehensive interpretation of vertex similarity. The considered constraints are also not time varying and so we use Jaya optimization for optimization of these constraints. Then, evolutionary optimization algorithm is used for decision making purpose to analyze the links between nodes. The proposed method has been tested on five different social networks such as AdjNoun, Anna Karenina, movie tweeting, UK faculty, and work place.

4.2. Proposed Model of Proposed Work

The model of proposed MCD method is illustrated in Figure 1. The working function depends on proposed evolutionary combined Jaya optimization algorithm. The social networks are taken as input dataset of proposed method which have been created in Network Simulator (NS-2) tool. The different constraints are computed and optimized by Jaya optimization algorithm. Then the link analysis is performed by evolutionary based decision making algorithm, which computes whether the node is attached to one or multiple nodes. Based on the associations between nodes and social objects, we partition the members involved in social objects into different groups.

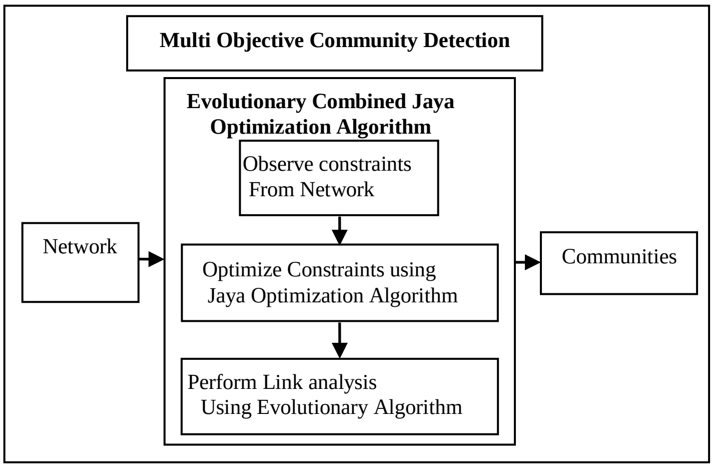

Multi Objective Community Detection
Evolutionary Combined Jaya Optimization Algorithm
Observe constraints From Network
Network
Optimize Constraints using Jaya Optimization Algorithm
Communities
Perform Link analysis Using Evolutionary Algorithm

Figure 1. Proposed Model of Proposed Multi Objective Community Detection (MCD) Method

5. Proposed Multi Objective Community Detection (MCD) Method

The detailed framework of proposed MCD method is illustrated in this section. Objective Functions have been discussed in Section 5.1. The constraint optimization has been discussed in Section 5.2. The link analysis present in Section 5.3.

5.1. Metrics for Community Detection

5.1.1 Modularity: Modularity of undirected with the un-weighted network is defined as the ratio of the difference between the actual and expected number of edges within the community. Modularity [23] measures the positive effect of grouping nodes together in terms of taking into account existing edges between nodes. Consider an undirected network $G=(V,E)$ with $|E|$ edges and modularity (Q) is given by

$$Q=\sum_{c_i \in C}\left[\frac{\left|E_{c_i}^{in}\right|}{|E|}-\left(\frac{2\left|E_{c_i}^{in}\right|+\left|E_{c_i}^{out}\right|}{2|E|}\right)^2\right] \tag{1}$$

where C is the set of all the communities, c_i is a specific community in C, $\left|E_{c_i}^{in}\right|$ is the number of edges between nodes within the community c_i, and $\left|E_{c_i}^{out}\right|$ is the number of edges from the nodes in the community c_i to the nodes outside c_i. Modularity of directed network is given as follows:

$$Q=\sum_{c_i \in C}\left[\frac{\left|E_{c_i}^{in}\right|}{|E|}-\frac{\left(\left|E_{c_i}^{in}\right|+\left|E_{out,c_i}\right|\right)\left(\left|E_{c_i}^{in}\right|+\left|E_{c_i,out_i}\right|\right)}{|E|}\right] \tag{2}$$

where $\left|E_{out,c_i}\right|$ is the number of edges from the nodes outside the community c_i to the nodes in community c_i and $\left|E_{c_i,out_i}\right|$ is the number of edges from the nodes in the community c_i to the nodes outside c_i. For undirected networks, it is clear that $\left|E_{out,c_i}\right|=\left|E_{out,c_i}\right|=\left|E_{c_i}^{out}\right|$ and thus the directed modularity is reduced to undirected modularity. But modularity maximization tends to support small communities as compared to larger one in some cases and sometimes larger communities as compared to smaller one in other cases. This tendency of modularity for preferring larger communities as compared to smaller ones is called the resolution limit problem. So, the modularity with a split penalty [24] (Q_{SP}) is compute by subtracting modularity from the split penalty (S_p)which is the fraction of edges that connect nodes of different communities. The S_p measures the negative effect of ignoring edges joining members of different communities. Enlarging community eliminates some S_p but if there are only a few edges across current partition, modularity of the merged community could be lower, negating the benefit of merging. Splitting a community into two or more communities introduces some S_p but if there are only a few edges between those separated communities, an increase of modularity can make such splitting beneficial. The S_p of the undirected network is given as follows:

$$S_p = \sum_{c_i \in C} \left[\sum_{\substack{c_j \in C \\ c_j \neq c_i}} \frac{\left|E_{c_i,c_j}\right|}{2|E|} \right] \tag{3}$$

where $\left|E_{c_i,c_j}\right|$ is the number of edges from community c_i to community c_j for unweighted networks or the sum of the weights of the edges for weighted networks. The S_p of directed network is given by

$$S_p = \sum_{c_i \in C} \left[\sum_{\substack{c_j \in C \\ c_j \neq c_i}} \frac{\left|E_{c_i,c_j}\right|}{|E|} \right] \tag{4}$$

The modularity with a split penalty (Q_{SP}) of undirected and directed networks expressed in equation (5) and (6) respectively.

$$Q_{SP-UD} = \sum_{c_i \in C} \left[\frac{\left|E_{c_i}^{in}\right|}{|E|} - \left(\frac{2\left|E_{c_i}^{in}\right| + \left|E_{c_i}^{out}\right|}{2|E|} \right)^2 - \sum_{\substack{c_j \in C \\ c_j \neq c_i}} \frac{\left|E_{c_i,c_j}\right|}{2|E|} \right] \tag{5}$$

$$Q_{SP-D} = \sum_{c_i \in C} \left[\frac{\left|E_{c_i}^{in}\right|}{|E|} - \frac{\left(\left|E_{c_i}^{in}\right| + \left|E_{out,c_i}\right|\right)\left(\left|E_{c_i}^{in}\right| + \left|E_{out,c_i}\right|\right)}{|E|} - \sum_{\substack{c_j \in C \\ c_j \neq c_i}} \frac{\left|E_{c_i,c_j}\right|}{2|E|} \right] \tag{6}$$

Modularity is not competent in finding valid communities, as the size of a detected community relies on the size of the whole network. The real reason is that the modularity does not include any information of the number of nodes in a community, and the detected community is highly sensitive to the total number of edges in the complex network. So, M. Chen *et al.*, [24] proposed a new metric called modularity density which can solve the problems associated with modularity. Modularity density $(\mathrm{Q_D})$ is computed form the equation (5) and (6) and $\mathrm{Q_D}$ of undirected and directed networks expressed in equation (7) and (8) respectively.

$$Q_D = \sum_{c_i \in C} \left[\frac{\left|E_{c_i}^{in}\right|}{|E|} d_{c_i} - \left(\frac{2\left|E_{c_i}^{in}\right| + \left|E_{c_i}^{out}\right|}{2|E|} d_{c_i} \right)^2 - \sum_{\substack{c_j \in C \\ c_j \neq c_i}} \frac{\left|E_{c_i,c_j}\right|}{2|E|} d_{c_i,c_j} \right] \tag{7}$$

$$Q_D = \sum_{c_i \in C} \left[\frac{\left|E_{c_i}^{in}\right|}{|E|} d_{c_i} - \frac{\left(\left|E_{c_i}^{in}\right| + \left|E_{out,c_i}\right|\right)\left(\left|E_{c_i}^{in}\right| + \left|E_{out,c_i}\right|\right)}{|E|} d_{c_i} - \sum_{\substack{c_j \in C \\ c_j \neq c_i}} \frac{\left|E_{c_i,c_j}\right|}{2|E|} d_{c_i,c_j} \right] \tag{8}$$

where d_{c_i} is the internal density of community c_i *i.e.,* $d_{c_i} = \dfrac{2\lambda}{|c_i|(|c_i - 1|)}$ and the d_{c_i,c_j} is the pair-wise density between community c_i and community c_j *i.e.,* $d_{c_i,c_j} = \dfrac{2(1-\lambda)}{|c_i||c_i|}$.

Note that λ is un-weighted for both un-weighted and weighted networks and it is always $0 \le \lambda \le 1$. The analysis of the topological structure reveals more detailed and hierarchical organization of complex network by tuning parameter λ.

5.1.2. Generalized Modularity: Generalized modularity is basically an extension of modularity which considers vertex similarity. Generalized modularity [25] is able to detect communities in unipartite, bipartite and near-bipartite networks without any pre-knowledge about the structure of network. It is inspired from the idea of modularity. It assumes that neighbors should lie in the same community. Likewise, two vertices are more likely to be in same community if they have more paths of length three or more, than the corresponding expected number in a randomize network. Generalized modularity takes the advantage of both vertex similarity and longer paths which are between the vertices. The generalized modularity is denotes as follows:

$$Q_g = \sum_{c_i, c_j \in V} c_1 \frac{N_p^{(l)} - \left(\left| E_{c_i}^{in} \right| + \left| E_{c_i}^{out} \right| \right)}{\left\| N_p^{(l)} \right\|} \tag{9}$$

where $N_p^{(l)}$ is the generalized modularity matrix, which represents the number of simple paths of length (l) between vertices. c_1 represents the weight of contribution of modularity matrix norms. The matrix of $N_p^{(l)}$ is equal to the adjacency matrix of simple paths of length l, for l = 1, 2. $\left\| N_p^{(l)} \right\|$ is the entry wise 1-norm of matrix $N_p^{(l)}$which is summation of absolute values of the matrix elements. E_{ci}^{in} and E_{ci}^{out}represents the expected number of in and out paths of length *l* between vertices in randomized network.

5.1.3. The Similarity of Attributes (SA): The similarity of attributes (SA) [22] of the node is calculated as follows:

$$SA = \frac{1}{d} \sum_{i=1}^{d} \max_{\substack{1 \le C_{j_1} \le K_1 \\ 1 \le C_{j_2} \le K_2 \\ \vdots \\ 1 \le C_{jA} \le K_A}} \left\{ \frac{\left(\sum_{h=1}^{A} n_{c_i c_{j_h}} \right)}{A \times n} \right\} \tag{10}$$

where d represents the number of detected communities, A represents the number of attributes for each node of the corresponding network, $n_{c_i c_{j_h}}$ refers to the number of nodes which their h-th attribute, N refers to the number of nodes in community, K_1 is the number of distinct values for the first attribute of nodes, and K_A is the number of distinct values for the Ath attribute of nodes. The higher value of SA means better communities partitioned on the basis of attribute similarity.

Modularity, generalized modularity, and similarity of attributes are the three objective functions which have been considered for optimization in our proposed work.

5.2. Constraint Optimization using Jaya Optimization Algorithm

The JAYA optimization algorithm begins with the initialization process in which we define the population size, design variables, and termination. Then compute the values of goal functions for the initial status, which modularity, generalized modularity, and similarity of attributes in equations (8), (9), and (10) respectively. Then create an initial random population based on the defined controlling parameters within the pre-specified limits of design variables. The population P(y) is formulated as follows:

$$\mathrm{P}(y)=\begin{bmatrix} p_1(1) & p_1(2) & \cdots & p_1(n) \\ p_2(1) & p_2(2) & \cdots & p_2(n) \\ \vdots & \vdots & \vdots & \vdots \\ p_m(1) & p_m(2) & \cdots & p_m(n) \end{bmatrix} \tag{11}$$

where n is the number of control variables, whereas m is the number of cluster members in the network of any iteration i. Let the best candidate best obtains the best value of $P(1)_b$ in the entire candidate solutions and the worst candidate worst obtains the worst value of $P(1)_w$ in the entire candidate solutions. If $P_{m,n,i}$ is the value of $\mathrm{n-th}$ variable for the $\mathrm{m-th}$ candidate during $\mathrm{i-th}$ iteration can be written as follows:

$$P'_{m,n,i}=P_{m,n,i}+r_{m,1,i}\left(P_{m,b,i}-\left|P_{m,n,i}\right|\right)-r_{m,2,i}\left(P_{m,w,i}-\left|P_{m,n,i}\right|\right) \tag{12}$$

where $r_{m,1,i}$ and $r_{m,2,i}$ is the random numbers between 0 to 1.

For example, consider the AdjNoun data set and it contains 112 nodes, *i.e.*, m=112, and design variables are three constraints *i.e.*, n=3. The modified population can be described as follows:

$$\mathrm{P}=\begin{bmatrix} p_1(1) & p_1(2) & p_1(3) \\ p_2(1) & p_2(2) & p_2(3) \\ \vdots & \vdots & \vdots \\ p_{112}(1) & p_{112}(2) & p_{112}(3) \end{bmatrix} \tag{13}$$

Run control stream program for every contender approach and figure the estimation of target cutoff that relates to each philosophy. See the best and most detectably accursed game-plans among the contender plots. In light of the best and most unmistakably dazzling diagrams, change each and every without question approach. For every restored plan, if any design variable upper most cleared point is hurt, supplant the evaluated a focal reason with past what many would consider conceivable. For each iteration, look at the target work values for the past and upheld approach. See the maintained layout in the event that it is better than the past value. Something else, keep the past value. Stop and report the ideal system if the end presentation is refined. Here, we use Rastrigin's function for fitness computation and define as follows:

$$f\,(i,j)=\sum_{i=1}^{n}\left[p_i^2-10\cos(2\pi\,\mathrm{p_i})+10\right] \tag{14}$$

The working flow of Jaya optimization is present in Algorithm 1. The optimized constraints are forward to the next link analysis phase to compute the exact similarities of each node using evolutionary based decision making algorithm.

Algorithm 1: Constraint Optimization using Jaya Optimization Algorithm

Input: Population size, number of design variables and termination criterion
Output: Optimized constraints

```
1.  Initialize the modularity, generalized modularity, and attribute of similarity.
2.  Compute initial solution using equation (14), and identify best and worst solution in the
        population.
3.  Modify the population solution using equation (12).
4.  Update the new solution if is better than old, otherwise maintain old one. The solution used
        to finalize the vehicle nodes are in the cluster or not.
5.  Stop the process if termination reached.
6.  Return: Optimized constraints.
```

5.3. Link Analysis using Evolutionary based Decision Making Algorithm

The problem with two or three objective function is said to be multi-objective problem, here, we gathered three information's from social networks and optimized them using Jaya optimization algorithm. The multi-objective problems are effectively solved by Evolutionary methods, which computes set of optimal solution. Multi objective evolutionary algorithm can be used to compute the optimized solution for real-world problems. We enhance the conventional algorithm by involving multi-ruled structure. This algorithm is used as a decision maker, which computes the link weights based on the optimized constraints. The modification is done by the special rules and evaluates the final solution to overcome problems. An evolutionary algorithm decomposes a number of single-objective optimization problems formulated by Rastrigin's function $(f_{i,j})$ using uniformly distributed weight vectors $p_i\ (i=1,2,\ldots,n)$. Each weight vector computes the unique search direction in the m-dimensional objective space with set of uniformly distributed weights. This method is used to avoid duplicate solutions and not predetermine target solutions to be updated before each iteration is generated and evaluated. The evolutionary based decision making algorithm computes the target solutions based on the different constraints with their location. The evolutionary based decision making algorithm updates the solution every second to reutilize the existing solution, called optimal updating. The algorithm starts with the initialization step. The objective possible paths are $p_i = (p_1, p_2, \ldots, p_n)$ of the objective function values. Rastrigin's functions which decompose the objective searching space based on the current ideal point I_i, for each link computes the weight value p_i as follows:

$$p_i = \frac{p_i(y_j) - I_i}{\sum_{l=1}^{m}\left(p_l(y_j) - I_l\right)}; \qquad j = 1,2,\ldots,d \tag{15}$$

where d is the number of constraints used for this optimization and the function of path1, $i=0$ (for example) as follows:

$$p_i(y_j) = p_1(y_1) + p_1(y_2) + p_1(y_3); \qquad i = 1, j = 3 \tag{16}$$

For each path $i \in (1, 2, \ldots, n)$, compute above three constraints between the source to destination position and store them as the corresponding distances (D). After computing the new weight value, compare the new value with old one. If the new weight value is best then update the new value, otherwise, maintain the old value. The optimal update method performs very well to update every seconds, which computes similarities in deep. The working flow of multi objective evolutionary algorithm is present in Algorithm 2.

Algorithm 2: Link Analysis using Multi Objective Evolutionary Algorithm

```
Input: Optimized constraints
Output: Link weight

1.   Initialize the optimized constraints.
2.   Compute the link weight using equation (15).
3.   Update the new solution if better than old, otherwise maintain old one.
4.   Return: Optimal link weights
```

6. Simulation Results and Observation

The proposed multi objective Community Detection (MCD) method is implemented in Network Simulator (NS-2) for social network in a 1000x1000 m^2 environment with the simulation time of 200 seconds. The simulated traffic is constant bit rate (CBR). In order to evaluate the performance analysis of the proposed MCD method is compared with the existing multi-objective discrete biogeography based optimization (BBO) [21], modularity, SimAtt maximization BBO Algorithm [22] in terms of α_{SA}. The metric used to compute the best compromise solution among set of non-dominated ones considers two aspects of node: attributes and link weights. It means that the metric should be able to find a solution in which members share similar attributes and have dense connections. The metric is defined as follows:

$$\alpha_{SA} = \frac{(1+\alpha^2)(SA \times Q_D)}{(\alpha^2 \times SA + Q_D)} \tag{17}$$

where α is a tuning factor between [0, ∞], which used to adjust the weight of link constraints. The proposed method has been tested with five different social networks such as AdjNoun, Anna Karenina, movie tweeting, UK faculty, and work place. The simulation parameters and data sets are given in Table 1 and 2 respectively.

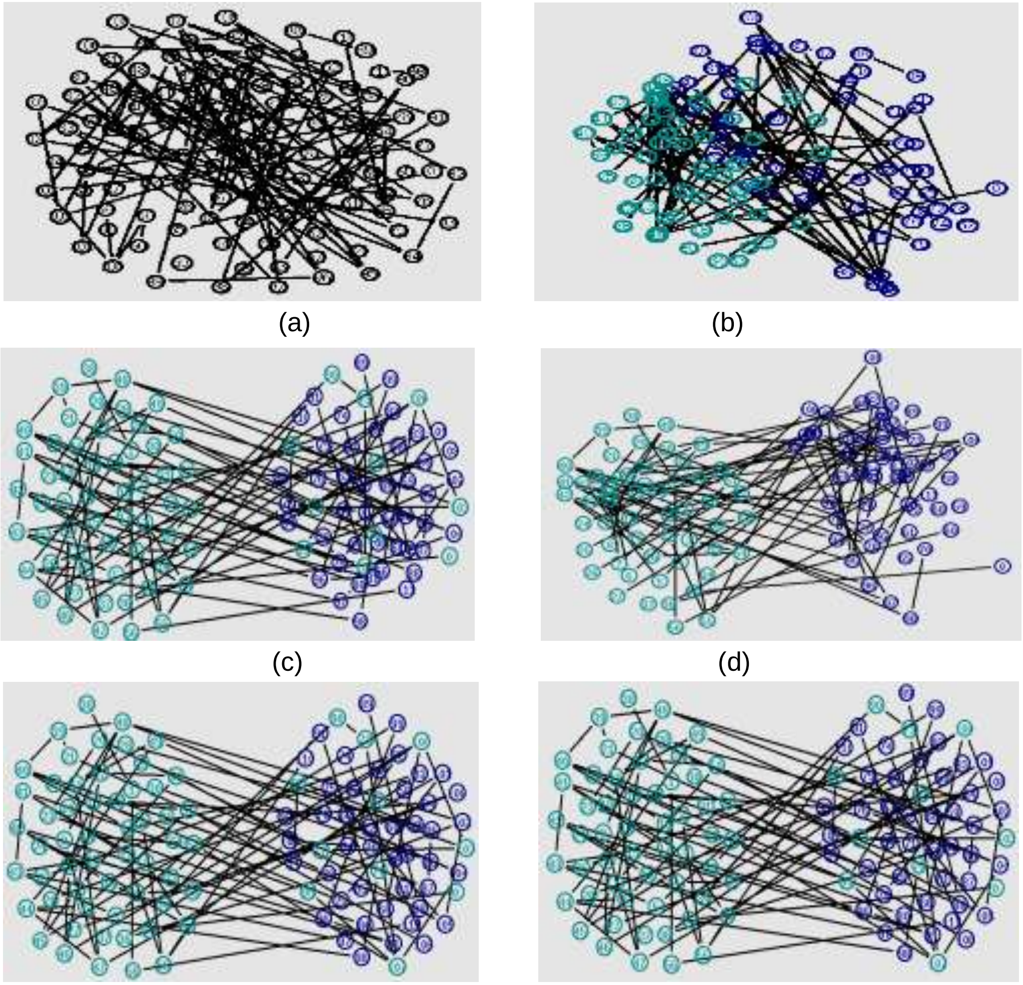
Table 1. Simulation Parameters

Parameters	Values
Node mobility	10 m/s
Network area	1000 X 1000 m^2
MAC	IEEE 802.11
Transmission range	250m
Traffic source	Constant bit rate (CBR)
Packet size	1024 bytes
Rate	50Kbps
Simulation time	500 sec

Table 2. Data Set for Social Networks

Network structure	AdjNoun	Anna Karenina	Movie tweeting	UK faculty	work place
Number of nodes	112	138	183	81	92

The data set contains the network of common adjective and noun adjacencies for the novel and nodes represent the most commonly occurring adjectives and nouns. In AdjNoun data set, node attributes are 0 for “adjectives” and 1 for “nouns”. Edges connect any pair of words that occur in adjacent position in the text of the book. The network of this data set contains 112 nodes. The output Nam for AdjNoun at different simulation time is given in Figure 2. Our proposed MCD method detects community by 2 groups as Adjectives and noun. Figure 2a shows the community detection network at starting stage. Figure 2b shows the community detection network of at 20th sec. simulation time. Figure 2c shows the community detection of network at 40th sec. simulation time. Figure 2d shows the community detection of network at 60th sec. simulation time. Figure 2e shows the community detection of network at 80th sec. simulation time. Figure 2f shows the community detection of network at 100th sec. simulation time. Community detection of network at 100th sec. simulation time. Similarly, we perform for rest four networks.



(a) (b)

(c) (d)

(e) (f)

Figure 2. Output Nam of AdjNoun at (a) t=0 sec (b) t=20 sec (c) t=40 sec (d) t=60 sec (e) t=80 sec and (f) t=100 sec of Simulation Time

6.2. Performance Comparison

The performance of proposed MCD method has been compared with the existing methods in terms of modularity, attributes similarity, and α_{SA}. The result shown in Tables 3 and 4 demonstrates that the modularity and attribute similarities are better than existing method. The proposed method has gained the highest mean values among all of the competitors on each data set. On the other hand, the performance of the proposed method considering the value of modularity outperforms the other methods in most of the times and equals to their results in other times. The reason for these better performances of the proposed method is that it considers modularity and optimizes it along with attribute similarities as a multi-objective task.

Table 3. Resultant Modularity Comparison of Proposed and Existing Methods

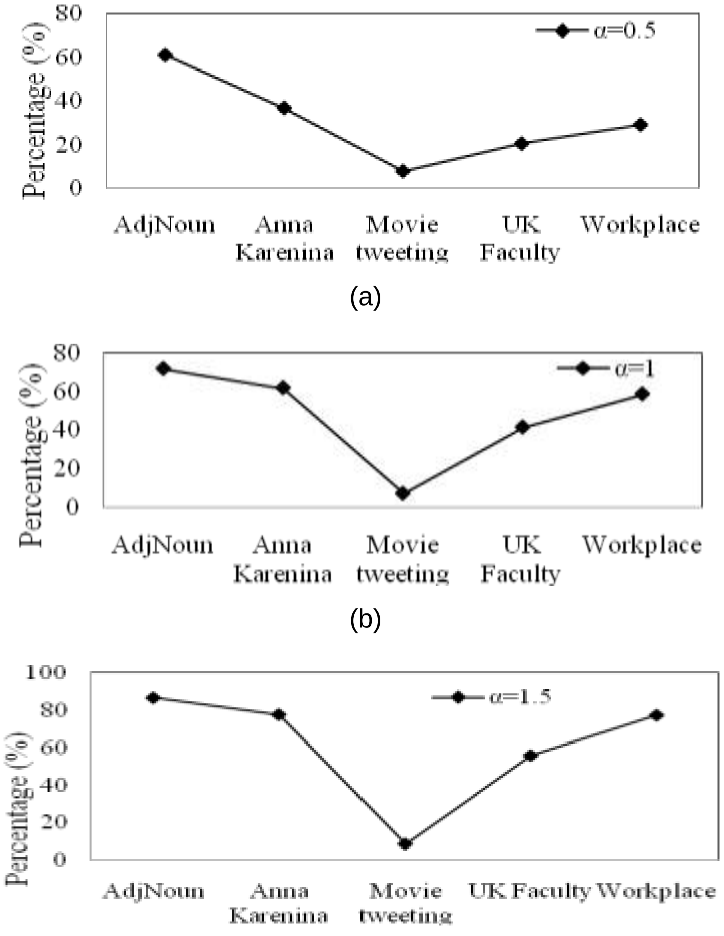
Data sets	Modularity	Modularity max. BBO [22]	SA max. BBO [22]	Multi-objective discrete BBO [21]	MCD method
AdjNoun	Mean	0.29304	0.02643	0.30434	0.37667
	Std	0.00743	0.00593	0.00319	0.00188
Anna Karenina	Mean	0.39512	0.05404	0.40027	0.46333
	Std	0.00237	0.00994	0.00393	0.00232
Moving tweeting	Mean	0.71178	0.38694	0.71306	0.61000
	Std	0.00157	0.03763	0.0041	0.00305
UK Faculty	Mean	0.44140	0.23568	0.4481	0.47000
	Std	0.00381	0.1968	0.00000	0.00135
Work place	Mean	0.37478	0.08351	0.38570	0.40666
	Std	0.01253	0.03435	0.00000	0.00153

This method considers modularity and optimizes it along with similarity attributes as a multi-objective task for final solution.

The resultant metric α_{SA} is used to determine the best compromise solution and the result values are tabulated in Table 5. The percentage increment of α_{SA} of proposed MCD method is better than existing multi-objective BBO method. The metric α_{SA} is computed for different α values as 0.5, 1, and 1.5. The performance comparison of metric α_{SA} with α=0.5, 1, and 1.5 is shown in Figure 3a-c respectively.

Table 4. Resultant Similarity Attributes Comparison of Proposed and Existing Methods

Data sets	SA	Modularity max. BBO [22]	SA max. BBO [22]	Multi Objective Discrete BBO [21]	MCD method
Adj Noun	Mean	0.58228	0.93231	0.97169	0.982609
	Std	0.03474	0.01140	0.00160	0.001130
Anna Karenina	Mean	0.68666	0.97085	0.98795	0.990734
	Std	0.02509	0.00380	0.00148	0.001390
Moving tweeting	Mean	0.94510	0.97449	0.98275	0.994505
	Std	0.00236	0.00118	0.00359	0.001830
UK Faculty	Mean	0.93847	0.99255	1.00000	0.704348
	Std	0.03623	0.00211	0.00000	0.000810
Work place	Mean	0.91041	0.95903	0.99804	1.000000
	Std	0.04754	0.00662	0.00000	0.000920



(a)

(b)

(c)

Figure 3. Metric α_{SA} Improvement of Proposed MCD Method Compare to Existing Multi-Objective BBO Method with (a) α=0.5 (b) α=1 (c) α=1.5

Table 5. Resultant α_{SA} Comparison of Proposed and Existing Methods

Data sets	α_{SA}	Modularity max. BBO [22]	SA max. BBO [22]	Multi- objective Discrete BBO [21]	MCD method
Adj Noun	0.5	0.48580	0.11820	0.59831	0.964752
	1	0.38949	0.05134	0.42296	0.868277
	1.5	0.34568	0.03768	0.36265	0.836118
Anna Karenina	0.5	0.59817	0.21992	0.70488	0.966140
	1	0.50143	0.10222	0.53506	0.869526
	1.5	0.45439	0.07612	0.46873	0.837322
Moving tweeting	0.5	0.88695	0.74588	0.89540	0.967596
	1	0.81201	0.55288	0.81358	0.870836
	1.5	0.77029	0.47450	0.77150	0.838583
UK Faculty	0.5	0.76574	0.60332	0.79713	0.961832
	1	0.60024	0.38050	0.61180	0.865649
	1.5	0.52724	0.30774	0.53431	0.833588
Work place	0.5	0.70773	0.30021	0.74486	0.963063
	1	0.53078	0.15180	0.54590	0.866756
	1.5	0.45752	0.11545	0.46977	0.834654

7. Conclusion

In this paper, we have proposed the Multi objective community detection (MCD) method for social networks. In MCD method, Jaya optimization algorithm can be used to optimize the time varying community detection metrics. These optimized values are then utilized to compute the link analysis using multi objective evolutionary based decision making algorithm. The proposed MCD method has been tested in five different social networks and the results have been compared with existing methods in terms of modularity, similarity attributes, and metric α_{SA}.

References

[1] G. Xu, S. Tsoka and L. Papageorgiou, "Finding community structures in complex networks using mixed integer optimization", The European Physical Journal B, vol. 60, no. 2, **(2007)**, pp. 231-239.

[2] C. Lee and P. Cunningham, "Community detection: effective evaluation on large social networks", Journal of Complex Networks, vol. 2, no. 1, **(2013)**, pp. 19-37.

[3] J. Kumpula, J. Saramäki, K. Kaski and J. Kertész, "Limited resolution in complex network community detection with Potts model approach", The European Physical Journal B, vol. 56, no. 1, **(2007)**, pp. 41-45.

[4] W. Pan, B. Li, Y. Ma, J. Liu and Y. Qin, "Class structure refactoring of object-oriented softwares using community detection in dependency networks", Frontiers of Computer Science in China, vol. 3, no. 3, **(2009)**, pp. 396-404.

[5] J. Liu and G. Deng, "Detecting community structure in networks by representative energy", Frontiers of Computer Science in China, vol. 3, no. 3, **(2009)**, pp. 366-372.

[6] F. Wei, W. Qian, C. Wang and A. Zhou, "Detecting Overlapping Community Structures in Networks", World Wide Web, vol. 12, no. 2, **(2009)**, pp. 235-261.

[7] M. Mørup and M. Schmidt, "Bayesian Community Detection", Neural Computation, vol. 24, no. 9, **(2012)**, pp. 2434-2456.

[8] L. Yu, B. Wu and B. Wang, "LBLP: link-clustering-based approach for overlapping community detection", Tsinghua Science and Technology, vol. 18, no. 4, **(2013)**, pp. 387-397.

[9] E. Gregori, L. Lenzini and S. Mainardi, "Parallel k-Clique Community Detection on Large-Scale Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 8, **(2013)**, pp. 1651-1660.

[10] F. Folino and C. Pizzuti "An Evolutionary Multiobjective Approach for Community Discovery in Dynamic Networks", IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 8, **(2014)**, pp. 1838-1852.

[11] S. Jia, Y. Gao, L. Gao and H. Wang, "Anti-triangle centrality-based community detection in complex networks", IET Systems Biology, vol. 8, no. 3, **(2014)**, pp. 116-125.

[12] F. Hao, G. Min, Z. Pei, D. S. Park and L. T. Yang, "K-Clique Community Detection in Social Networks Based on Formal Concept Analysis", IEEE Systems Journal, vol. 11, no. 1, **(2017)** March, pp. 250-259.

[13] X. Ma and D. Dong, "Evolutionary Nonnegative Matrix Factorization Algorithms for Community Detection in Dynamic Networks", IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 5, **(2017)**, pp. 1045-1058.

[14] J. Wang and I. Paschalidis "Botnet Detection Based on Anomaly and Community Detection", IEEE Transactions on Control of Network Systems, vol. 4, no. 2, **(2017)**, pp. 392-404.

[15] F. Huang and X. Li, "Overlapping Community Detection for multimedia social networks", IEEE Transactions on multimedia, vol. 19, no. 8, **(2017)** August.

[16] A. Mahmood, M. Small, S. Al-Maadeed and N. Rajpoot, "Using Geodesic Space Density Gradients for Network Community Detection", IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 4, **(2017)**, pp. 921-935.

[17] K. Žalik, "Community detection in networks using new update rules for label propagation", Computing, vol. 99, no. 7, **(2016)**, pp. 679-700.

[18] R. Interdonato, A. Tagarelli, D. Ienco, A. Sallaberry and P. Poncelet, "Local community detection in multilayer networks", Data Mining and Knowledge Discovery, **(2017)**.

[19] A. Tagarelli, A. Amelio and F. Gullo, "Ensemble-based community detection in multilayer networks", Data Mining and Knowledge Discovery, **(2017)**.

[20] Z. Li, J. Chen, Y. Fu, G. Hu, Z. Pan and L. Zhang, "Community Detection Based on Regularized Semi-Nonnegative Matrix Tri-Factorization in Signed Networks", Mobile Networks and Applications, **(2017).**

[21] A. Reihanian, M. Feizi-Derakhshi and H. Aghdasi, "Community detection in social networks with node attributes based on multi-objective biogeography based optimization", Engineering Applications of Artificial Intelligence, vol. 62, **(2017)**, pp. 51-67.

[22] X. Zhou, Y. Liu, B. Li and G. Sun, "Multiobjective biogeography based optimization algorithm with decomposition for community detection in dynamic networks", Physica A: Statistical Mechanics and its Applications vol. 436, **(2015)**, pp. 430-442.

[23] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks", Physical Review E, vol. 69, no. 2, **(2004)**.

[24] M. Chen, T. Nguyen, and B. K. Szymanski, "A new metric for quality of network community structure", ASE Human Journal, vol. 2, no. 4, **(2013)**, pp. 226-240.

[25] M. Ganji, A. Seifi, H. Alizadeh, J. Bailey and P. J. Stuckey, "Generalized Modularity for Community Detection", In Machine Learning and Knowledge Discovery in Databases, Springer, **(2015)**, pp. 655-670.