Aspiration Criteria Based Graph Clustering with Greedy Initialization

Mousumi Dhara and K. K. Shukla

Department of Computer Engineering, IIT (BHU), Varanasi, India
mousumi.dhara@gmail.com, kkshukla.cse@itbhu.ac.in

Abstract

Clustering has an extensive and long history in a variety of scientific fields. Several recent studies of complex networks have suggested that the clustering analysis on networks has been an emerging research issue in data mining due to its variety of applications. Many graph clustering algorithms have been proposed in recent past, however, this clustering approach remains a challenging problem to solve real-world situation. In this work, we propose an aspiration criteria based graph clustering algorithm using stochastic local search for generating lower cost clustering results in terms of robustness and optimality for real-world complex network problems. In our proposed algorithm, all moves are meaningful and effective during the whole clustering process which indicates that moves are only accepted if the target node has neighbouring nodes in the destination cluster (moves to an empty cluster are the only exception to this instruction). An adaptive approach in our method is in incorporating the aspiration criteria for the best move (lower-cost changes) selection when the best non-tabu move involvements much higher cost compared to a tabued move then the tabued move is permitted otherwise the best non-tabu move is acceptable. Extensive experimentation with synthetic and real power-law distribution benchmark datasets show that our algorithm outperforms state-of-the-art graph clustering techniques on the basis of cost of clustering, cluster size, normalized mutual information (NMI) and modularity index of clustering results.

Keywords: *Cost of clustering, Cluster size, Normalized Mutual Information (NMI) and Modularity Index of Clustering Results, RNSC*

1. Introduction

Cluster analysis of graphs is a fundamental data analytic technique which explores mathematical modelling of diverse problems related to nature and society. For creating the graph clustering phenomena successful, different types of algorithms and methods have emerged over the years and these are often encountered into many applications such as data mining [1], machine learning [2], complex network analysis [3, 4], image segmentation [5, 6], information retrieval [7], bioinformatics [8, 9], study of social networks [10], continues to be employed in the field of sociology to explore social interactions, the study of biochemical networks [11, 12], biological neural networks [13, 14], and transport and communication networks. Although many successful graph clustering algorithms have been proposed in the recent decades, clustering is till now a tough problem. The notion of clusters is strongly dependent on the situation as well as the purpose of clustering to deal with all real-world cluster analysis problems. The essential problems [15] of a graph clustering algorithm are

identified and stated as follows. It is data driven to choose appropriate measure of similarity or dissimilarity between data points. It is a challenging task to select the best one among diverse measures. Considering the conceptual viewpoint of a cluster, there exist widely acceptable definitions of a cluster but it is difficult to develop strong fundamental framework of a concrete algorithm. However, these concepts are not capable of giving much impact to produce a robust and optimal algorithm. Optimality of an algorithm is mainly determined by the quality of clusters and it is computed by measuring the mutual information sharing between clusterings. The quality can be hampered by background noise or outliers. Therefore, it is necessary to eliminate or to distinguish them from actual clusters, to achieve optimal and robust clustering. Many of the clustering algorithms, which are built on the conception of minimizing the cost of error iteratively, suffer from getting trapped in local minima. The hyper parameters such as the kernel width in spectral clustering are not easy to tune manually [16]. An overview of graph-based clustering algorithms that are related to our work is stipulated herein. Widespread literature survey for clustering can be found in [17, 18]. Commonly, the abstract knowledge of graph based clustering algorithms comprises of three major steps.

(1) Create an underlying graph to derive a geometric structure among data points.

(2) Remove few inconsistent edges according to some important instructions.

(3) Recognize clusters from resulting subgraphs.

The basic concept of graph clustering is the separation of sparsely connected dense subgraphs from each other. In the recent past, various other graph clustering algorithms came into the field like Restricted Neighbourhood Search Clustering (RNSC) [19], Markov clustering (MCL) [20], Super Paramagnetic Clustering (SPC), Genetic Algorithm, Molecular Complex Detection (MCODE), Local Clique Merging Algorithm (LCMA), etc. RNSC, which is a cost based clustering method and performs local search iteratively to obtain optimum clustering in an efficient way. RNSC is a stochastic technique which uses restricted neighbourhood search concept. It acts like a metaheuristic technique as for example tabu search, described in [21] and can be used in various search space schematics. Tabu search concept was first proposed by Glover in [22] and is described in detail in [23]. It is a meta-heuristic, one that guides local search heuristics. The idea behind it is to allow cost-based local search algorithms to enter, then leave local minima by preventing the search from retracing its steps and settling in a local minimum. RNSC is also known as Variable neighbourhood search [24]. A restriction is imposed in the neighbourhood for the current clustering while doing iterative local search. The main goal of this algorithm is to find the best cost clusterings (lower cost) from the set of clusterings of a graph by assigning some cost functions (Naive cost function and scaled cost function). The memory requirement for RNSC is O (n^2). The complexity of a move in the naive cost function is O (n), which is the size of the restricted neighbourhood of a move M. A graph-clustering algorithm (MCL) incorporating the idea of performing a random walk on the graph to identify the more densely connected subgraphs is presented in van Dongen [20]. MCL is an efficient clustering method in weighted graphs, based on the prototype of stochastic flow simulation technique. In this technique, clusters (a natural grouping of densely flow-connected vertices) are obtained by using two operators: flow expansion and inflation. MCL technique performs well for sparse graphs. The expansion step of MCL has complexity O (n^3), assuming some small bound on the expansion exponents e_i. The inflation has complexity O (n^2). The idea of random walks is also used in [25], but only for clustering geometric data. Obviously, there is a close connection between graph clustering and the classical graph problem minimum cut. In the

case of weighted graphs, the simple paradigm gains additional ambiguities, namely, the interpretation of sparse, yet heavy, or dense, yet light, subgraphs. These potential groups fulfil the density or weight criterion, while failing the other. Thus their relevance as clusters is questionable or at least depends on the application. Along the lines of [26] using unweighted graphs, we concentrate on indices and algorithms that focus on the relation between the number of intracluster and intercluster edges. In [27] some indices measuring the quality of graph clustering are discussed. Conductance, an index concentrating on the intracluster edges is introduced and a clustering algorithm that repeatedly separates the graph is presented. A purely graph-theoretic approach using this connection, more or less directly, is the recursive minimum cut approach presented in [28]. Hartuv and Shamir, among others, proposed a clustering model based on high cluster connectivity [29]. Very recently, the physics community presented techniques based on centralities and statistical properties. For example, an algorithm that iteratively prunes edges based on betweenness centrality was introduced as a clustering technique in Newman and Girvan [30]. A related quality measure named *modularity* was presented in [31]. It evaluates the significance of clustering with respect to the graph structure by considering a random rewiring of the edge set. Nascimento and Eades applied simulated annealing to graph clustering, but the focus of their work was the integration of user participation in graph clustering algorithms [32]. In [33], Hoos and Stutzle compare systematic search algorithms with stochastic local search algorithms for 3-SAT, the propositional satisfiability problem with three literals in each clause.

Many complex systems in nature and society can be represented in terms of networks or graphs. Networks are universal in nature and society [34], defining various complex systems, such as the society, a network of individuals linked by various social links [35]; the Internet, a network of routers connected by various physical connections [36]; the world wide web, a virtual web of documents connected by uniform resource connectors [37] or the cell, a network of substrates connected by chemical reactions [38]. Particularly, it has been found that many networks of scientific interest are possessed scale-free property [39], that is, the probability that a randomly selected node has exactly k links decays as a power law, following

$$P(k) \sim k^{-\gamma}, \quad (1)$$ where γ is the degree exponent.

The list of recognized scale-free networks now consist of the world wide web, the Internet, the cell, the web of human sexual contacts, the language, or the web of actors in Hollywood, genetic networks, ecological networks, information engineering, citation networks, most of which appear to have degree exponents between two and three.

In this work, by introducing novel formulation of cost measurement we present an aspiration criterion based graph clustering algorithm that improves few objectives related to graph clustering. It is designed as an adaptive way that can alleviate the aforementioned problems. Mainly, the aim of this work is to improve the performance of RNSC algorithm through detection of its weaknesses. Performance evaluation of the proposed technique is processed using a range of synthetic and real power-law distribution benchmark dataset. The widespread experiments on these datasets demonstrate that the proposed technique is producing better clustering effectively in terms of robustness and optimality.

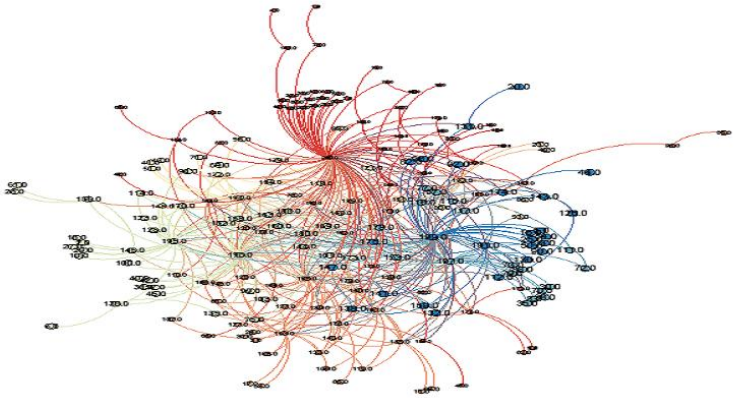

Figure 1. Power-law Graph with 200 Nodes

2. Background

2.1 Fundamentals of Restricted Neighborhood Search Clustering (RNSC)

RNSC is a local search meta-heuristic technique which is used to minimize the cost of clustering in the solution space. According to Stijn van Dongen, the vertex-wise performance criteria for clustering of unweighted graphs as the sum of the coverage measure taken on each vertex. In RNSC, a simple integer-valued cost function (called the naive cost function) is used as a pre-processor to produce initial clustering results on a graph and after that to evaluate the low-cost clustering result, a more expressive (but less efficient) real-valued cost function (called the scaled cost function) is applied. The scaled function tries to optimize the output from naive function and reach to the global optimal solution. For a clustering C on an un-weighted graph G (V, E) in which |V| = n, more expressive scaled coverage measure is in the following expression where, *N (v)* is the open neighbourhood of *v*.

$$Cov(G,C,v) = 1 - \frac{\neq^{1}_{out}(G,C,v) + \neq^{0}_{in}(G,C,v)}{N(v)C_v} \tag{2}$$

The scaled cost function is expressed as in Eq. (4).

$$C_s(G,C) = \frac{n-1}{3}\sum_{v\in V}\frac{1}{|N(v)\cup C_v|}(\neq^{1}_{out}(G,C,v) + \neq^{0}_{in}(G,C,v)) \tag{3}$$

Cost functions for weighted graphs: If *Wu, v* is the weight of the edge between vertices *u* and *v*. α_v is the cost numerator for *v* in a simple unweighted graph and that can be transformed to achieve γ_v, the cost numerator for *v* in a weighted graph. Define γ_v as follows.

$$\gamma_v = \sum_{u\notin C_v} w_{u,v} + \sum_{u\in C_v}(1 - w_{u,v}). \tag{4}$$

$$\beta_v = \sum_{v\in V}|N(v)\cup C_v| \tag{5}$$

With the new cost numerator γ_v defined in Eq. (4), the scaled cost function may be written as in Eq. (6).

$$C_s(G,C)=\frac{n-1}{3}\sum_{v\in V}\frac{\gamma_v}{\beta_v} \tag{6}$$

2.1.1 Limitations of RNSC

RNSC is an effective technique capable of solving many graphs clustering problems in the last decade. It provides suitable clusterings for some problems even better than MCL. But there are few fundamental weaknesses found out in this method. These limitations are divided into two categories; lack of effectiveness and fragmentation of output respectively.

2.1.1.1 Lack of Effectiveness

i) Complete re-generation of Candidate list.

ii) Moves for a node to cluster with no neighbouring nodes in it are also included. These moves are costlier than moving it to an empty cluster.

iii) The cost function evaluation in RNSC is not exact in their measure and a large number of moves are required during the computation, out of which few moves are meaningful.

iv) Cost scheme ignores the effect of a move on other nodes of the clusters involve on the move.

v) Costlier Scaled cost computation.

2.1.1.2 Fragmentation of Output

Tendency of RNSC is yielding too many clusters. For example, *S. cerevisiae* PPI network originating from Von Mering, *et al*., (2002) [40] comprising 11000 interactions with 2401 nodes (proteins) is tested on RNSC and 1155 clusters produced by RNSC in that case. From that scenario it is obvious that the method produces unnecessarily many clusters.

2.2 Aspiration Criteria

It is a rule for overriding the tabu restrictions at various levels to enhance the flexibility in tabu search. The tabu status of a solution is not an absolute. That can be overruled if certain conditions are encountered, conveyed in the form of aspiration levels. In effect, these aspiration levels deliver thresholds of attractiveness that direct whether the solutions may be considered admissible despite of being classified tabu. Clearly a solution better than any previously seen deserves to be considered admissible. The simplest and most commonly used aspiration criterion, found in almost all tabu search implementations, permits a tabu move when it results in a solution with an objective value better than that of the current best-known solution (since the new solution has obviously not been previously visited).The phenomenon behind using aspiration criterion is to improve tractability in the tabu search by leading it towards better moves. The overall description about the aspiration criteria is presented in the following Figure 2.

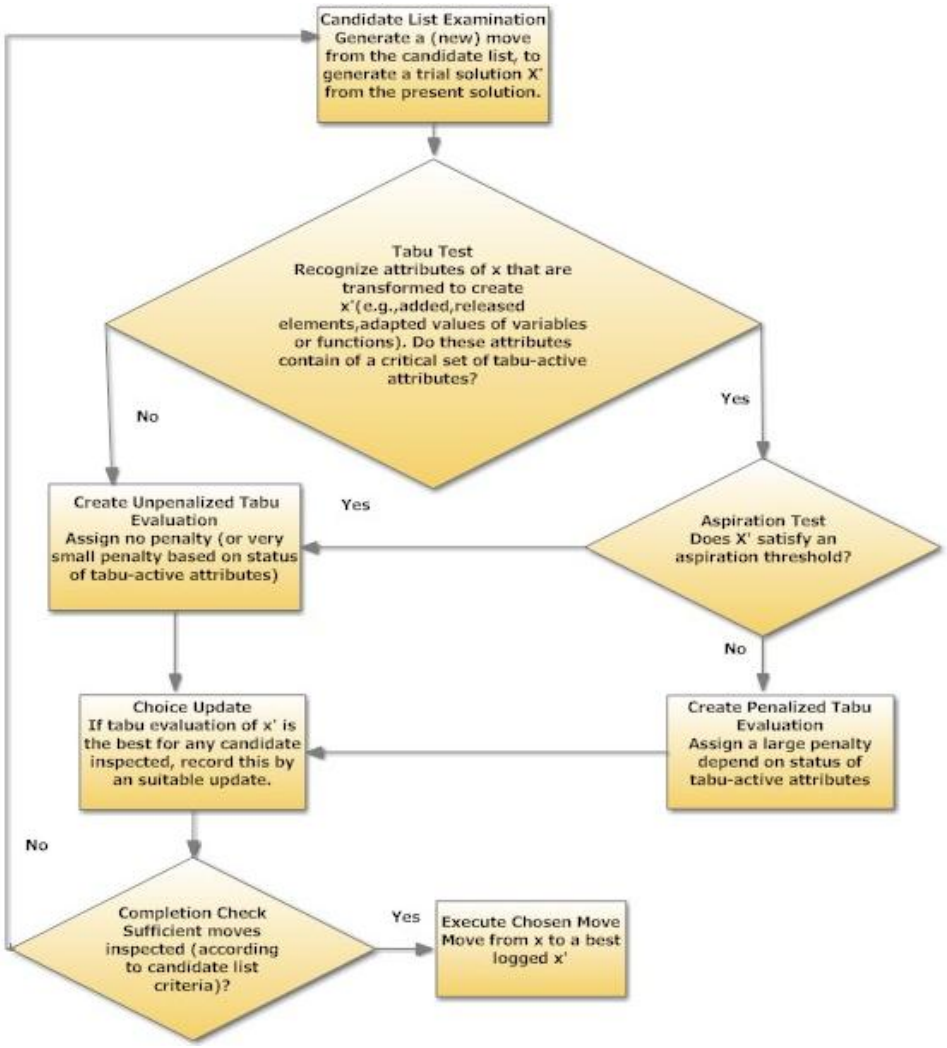

Figure 2. Short-term Memory based Tabu Evaluation

3. Description of Aspiration Criteria Based Graph Clustering algorithm (ACOGCT)

The proposed algorithm is developed by using advantage of the intellectual conception of tabu search. The main intension is to design a more significant and optimal algorithm for providing better clustering results by exploring some advanced concepts as aspiration criteria in tabu search. The step by step evaluation of our algorithm is deliberated below.

3.1 Overview of the Algorithm

In this section, summary of the steps of the developed algorithm is presented to acquire a quick insight into the logic involved.

Step 1

Create an initial clustering solution: This step involves assigning nodes to their cluster either on a random or on some other basis.

Step 2

Generate Move list: Generate a set of all possible moves and associate cost with them.

Step 3

Update Move list: Update the list of moves based on the last move. Last move may have brought changes in the cost of nodes in the move's source or destination cluster or both. They might be inclusion or exclusion of the moves.

Step 4

Move selection: Move may belong from the candidate list or be a diversification move.

Step 5

Apply the move: Update cluster and nodes about the application of the move. Save the best answer at local minima.

Step 6

Check: If the specified number of moves has not been applied, then jump to update move list.

Step 7

Return: Print the best answer and Exit.

3.2 Comparative Features of RNSC and Proposed Algorithm

The proposed algorithm is the refinement of RNSC with respect to few positive aspects. These features are conferred with proper explanations.

3.2.1 Key Positive Features

Few positive features are pointed out here to lay the foundation of the algorithm better compare to RNSC.

- Scale cost evaluation is O (n) in RNSC. This can easily be done in O (1) time if the information about current node, and its cluster contribution are pre-computed.
- RNSC might tabu some very good moves based on the tabu criteria. Instead, in the proposed algorithm, aspiration criteria serve the sole purpose of avoiding tabu (based on the relative cost of the best non-tabu move).
- Regeneration of all possible moves to select the best move, each time before it is applied in RNSC.
- Moves are considered only if the target node has neighbouring nodes in the destination cluster (moves to empty cluster are the only exception to this rule).
- The effect of a move for any cost scheme considered in RNSC is not exact in nature. They ignore the effect of moving on nodes other than the target node.
- Cost scheme is evaluated in RNSC on an absolute basis after each move. Instead, in the proposed algorithm, costs are evaluated relative to starting clustering state and iteratively. The cost of the starting cluster is set equal to zero and effects of moves are

added upon it. So, the effects of a move are added to get the cost of current clustering state relative to the initial clustering solution.

3.2.2 Features Retained in our Proposed Algorithm

The properties that are kept unchanged and taking advantage of this retained properties of RNSC is focused here.

- Short-term memory considerations using Tabu criteria are actively used.
- As in the case of RNSC, diversification moves are applied when in the recent past no good solution was found.
- Scale cost scheme forms the basis for evaluating the cost of a move.

3.3 Greedily Create an Initial Clustering Solution

There are different ways to perform the operation to **c**reate an initial clustering solution. Most common is the random clustering method that used in RNSC. Our algorithm uses a greedy initial clustering instead of random clustering. Due to this clustering, most of the nodes are placed such a way that there are good chances for some of its neighbours residing in the same cluster.

The initial clustering solution technique is explained here with the proper manner:

I. Select the node with the highest degree with no cluster assigned yet.

II. Add node to a new cluster and its unassigned neighbours are also put into the same cluster.

III. If all nodes haven't been assigned yet then go back to the initial step I.

3.4 Move selection

The idea behind the selection of a move similar to the technique used in RNSC, where type of move is decided based on the previous clustering costs or improvements. Diversification move is executed when there has been no improvement in the best cost of the clustering over the last specified interval of time otherwise a normal move (in our case tabu move) is applied. Diversification when run shuffles the current clustering by the specified amount of diversification period and frequency, even if it means a significant increase in the current cost. This helps us to get out of any local minima where we might have been stuck in and explore some new possible clusterings.

If there is no need for diversification, best move from the candidate list is selected if it's not on the tabu list (*i.e.*, the target node wasn't moved in the near past).

Our algorithm satisfies the aspiration criteria, whereas RNSC does not follow this criterion. Aspiration criteria allow selection of a move even if it's already tabued when the best non-tabu move incurs a cost which is much higher than itself. The basic idea is that, if the best move is already tabued instead of ignoring it, check the feasibility see if this move is going to be much better than the best non-tabu move existent. This difference between best move cost (which is in tabu) and best non-tabu move cost, if less than the aspiration level, then select the non-tabu move, otherwise select the best move.

3.5 Application of a MOVE

In this algorithm when a move is made then the target node is removed from the source cluster and added to the destination cluster. During execution of a move, a list of changes that contains the whole information about the source and target cluster is passed to each node related to those sources and destination cluster. Each node now quickly updates based on the changes it's going to incur the value for the total edge connections and edge weight with the neighbouring nodes in the cluster. These values later help in O (1) scale cost associated with the node.

After the updates on nodes and clusters performed tabu-list is informed about the changes that have occurred. Tabu list now identifies the last target node as tabu with duration depending on the previous tabu duration value associated with the target node. Greater the previous tabu duration value much greater will be the penalty added to the target node, so that the occurrence of moves with the node is forbidden.

3.6 Cost Estimation

Move stored in the candidate list other than consisting of a target node to be moved from the source cluster to destination cluster and also the recomputed cost is going to incur. The scaled cost scheme for weighted graph is described briefly in this section.

The scaled cost scheme: RNSC's scaled cost evaluation is costly due to the O (n) computation of a denominator value (β). In RNSC only the direct cost associated with the move is considered, *i.e.*, the changes in the cost for target node. It does not consider the effect induced on the nodes of the source and destination cluster.

In our algorithm, only the scaled cost evaluation is used. Scaled cost evaluated with any node could be computed in O (1) time against the O (n) time spent in the case of RNSC. The faster computations are due to constant update about the changes in the cluster to its node. Each node now quickly updates based on the changes it's going to incur the value for the total edge connections and edge weight with the neighbouring nodes in the cluster. The value of the total number of edge connections and total edge weights with the neighbouring nodes in the cluster have incurred during the node update process and based on that incurred value some changes are made for each node. This argument justifies that there is no need of using the naïve cost scheme. A cost change caused by the move is due to the sum of changes in the cost associated with the nodes of the source and the destination cluster. Direct cost is the change in the cost of a target node (moving node) itself. Induced cost is the sum of changes in cost of nodes belonging to the source or destination cluster other than the target node.

Logical view of cost changes with move evaluations:

In the new algorithm, only scale cost scheme from RNSC is used for move evaluations. Scale cost evaluations have been simplified to simple constant time operations, due to active update of information corresponding to nodes about its cluster contributions. Further, move evaluations have been broken down and simplified for a clearer understanding.

Let scale cost for node "t" in a cluster "c" is represented by Scale cost (t, c). For a graph with n vertices, Scale cost (t, c) ignores the constant multiplier of (n-1) /3 during the discussion ahead. Scale cost value combines results of contributions from interconnection,

intra-connection and neighbourhood (nodes present in the cluster c or have an edge with the node n).

α (numerator) = Weight due to inter-cluster connections of t+ weight due to intra-cluster connections of t.

β (denominator) = neighbourhood size.

$$\text{Scale Cost} = \alpha/\beta \tag{7}$$

Inter-Cluster contributions are the sum of all the connections from "t" to nodes in clusters other than "c". This adds the cost associated with inter-cluster connection, as they should have formed an intra-cluster connection.

Inter Cluster weight = (Total Edge weight of "t" – sum of all edge weights (of "t") within the cluster "c").

Intra-cluster contributions evaluate to a difference of maximum possible intra-connection value (edges formed with all vertices in the cluster) and the actual value.

Let M be the maximum possible edge weight and Nc be the size of the cluster "c".

Intra Cluster weight = (Nc-1) *M – sum of all edge weights (of "t") within the cluster "c".

Let total edge weight of t be represented is Wt. The total edge weight of connections or edges within the cluster c with one of its vertices being "t" and that is represented as Wc, t.

Number of edges of the node t is represented as Et.

Number of connections or edges within the cluster c with on its vertex being "t" and which is represented as Ec,t.

$$\alpha = (Wt - Wc,t) + (M*(N-1) - Wc,t) \tag{8}$$

$$\beta = Et + (Nc-1) - Ec,t \tag{9}$$

Since, α= Wc,t and Ec,t are constantly updated after application of each move, cost evaluation for any node in its current cluster is evaluated in constant time (will not be true if "c" doesn't contain "t").

Move consists of a target node (represented by "T"), source cluster (represented by "S") and the destination cluster (represented by "D"). On applying the move target node is moved from the source cluster to the destination cluster. Let there be a temporary empty cluster E.

Moving a node from a cluster to cluster brings changes in the costs associated with the nodes in the target and destination clusters also. This further impacts any move with its source or destination cluster equal to the last applied move's source or destination cluster. So for any move there are two costs associated.

(a) Direct Cost : Cost change on the target node "t"

(b) Induced Cost: Cost change on nodes in source & destination cluster other than "t".

Let E, be a temporary empty cluster.

The move is broken down into two simple steps.

(a) Move T from S to empty cluster E: mark all connections of node T as inter-cluster connection.

(b) Move T from E to destination D: unmakes connection to node T to nodes in the destination cluster as inter-cluster connection (intra-cluster connection).

Move Effect on cost = change in cost due to step (a) + change in cost due to step (b).
Remove Effect:

This step involves moving node T from S to E;

Let S' represent S after the move.

Direct-Remove-Effect = Scale cost (T, E) – Scale cost (T, S).

Induced-Remove-Effect as shown below contains changes in cost with other nodes.

For each node R in S {

Induced-Remove-Effect += (Scale cost (R, S') – Scale cost (R, S));

}

The new scaled cost evaluated from the induced-remove effect is measured by following two conditions. The conditions are stated as node in the source cluster was directly connected to the moved node and node in the source cluster wasn't directly connected to the moved node.

$$\text{Scaled cost}(R, S') = \left(\frac{\alpha_R + \Delta\alpha_R}{\beta_R + \Delta\beta_R}\right) \tag{10}$$

Where $\Delta\alpha_R = \begin{cases} -M + 2 \times \text{weight}(T,R); & \text{if connected} \\ -M + 2 \times \text{weight}(T,R); & \text{if not connected} \end{cases}$ and $\Delta\beta_R = \begin{cases} 0; & \text{if connected} \\ -1; & \text{if not connected} \end{cases}$

Total Remove Effect = T.R.E = Direct-Remove-Effect + Induced-Remove-Effect;

Add Effect:

This involves moving node from temporary empty cluster E to destination cluster D.

Let D', be the new state of D after a move has been applied.

Direct-Add-Effect = Scale cost (T, D) – Scale cost (T, E).

Induced cost is the sum of all cost changes on other nodes of the destination cluster.

For each Node R in D {

Induced-Add-Effect += (Scale cost (R, D') – Scale cost (R, D));

}

The new scaled cost, evaluated from the induced-add effect is measured by following two conditions. The conditions are specified as a node in the destination cluster was connected to the node added and a node in destination cluster wasn't directly connected to the moved node.

$$\text{Scaled cost (R, D')} = \left(\frac{\alpha_R + \Delta\alpha_R}{\beta_R + \Delta\beta_R}\right) \quad (11)$$

Where $\Delta\alpha_R = \begin{cases} \text{M-2}\times\text{weight (T,R); if connected} \\ \text{M-2}\times\text{weight (T,R); if not connected} \end{cases}$ and $\Delta\beta_R = \begin{cases} 0;\ \textit{if connected} \\ 1;\ \textit{if not connected} \end{cases}$

Total Add Effect = T.A.E =Induced-Remove-Effect + Induced-Add-Effect.

The details of adaptive scaled cost estimation and the proposed algorithm are stipulated in the Figure 3.

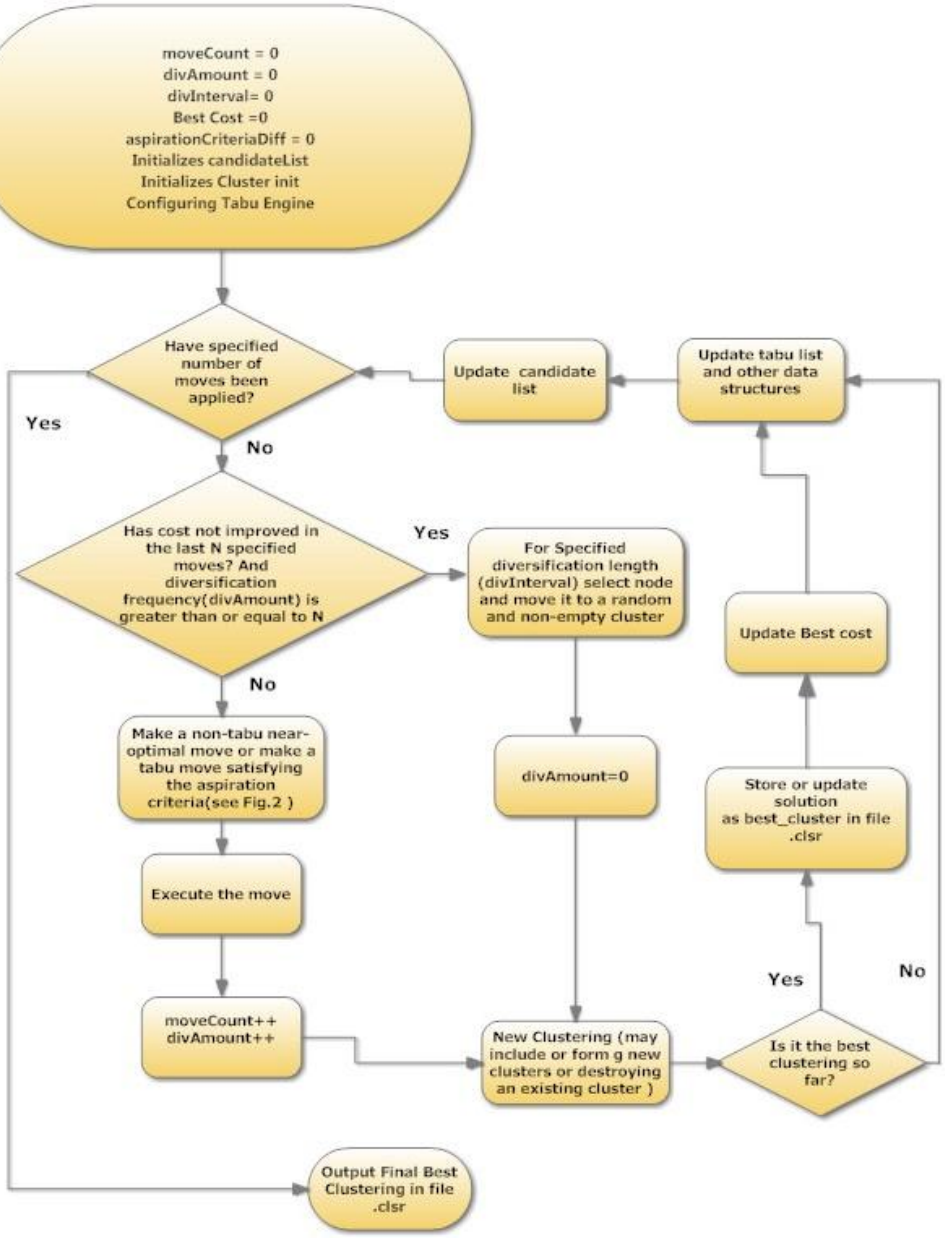

Figure 3. Adaptive Scaled Cost Evaluation

4. Experimental Results and Discussions

The evaluation of the performance in terms of robustness and quality of our proposed algorithm ACOGCT is compared with few selected state-of-art graph clustering algorithms as RNSC and MCL. The experiments are performed on a PC with a 2.53 GHz Intel (R) core (TM) 2 Duo and 2 GB of RAM. Some synthetic and real benchmark power-law distribution datasets are chosen to conduct the analysis of accuracy measure of graph clustering algorithms through computation of few performance metrics. The real power-law distribution networks are listed in table1 with respective references. We set up an initial configuration for creating the environment same for ACOGCT, RNSC and MCL to carry forward the experiments. The initial configuration for ACOGCT, RNSC and MCL is as follows. For ACOGCT, the number of moves denoted as move Count=1000; shuffling frequency denoted as div Amount =40; diversification length denoted as div Interval=10 and tabu-length=250. For RNSC, the following parameters are set like as d (diversification Length) = 10; D (shuffling Frequency) = 40; t (tabu-length) = 250 and e (number of experiments) = 1000 and in case of MCL, the inflation (I) value is 4; reweight loops c= 0. 25; pre-inflation value p= 0. 8 and preset resource scheme= 5.

Table 1. Real Power-law Network Data sets

Real Power-Law Networks	Graph Size	Average Degree <k>	Degree exponent (γ_{out})	Degree exponent (γ_{in})
Electronic circuits [41]	329	3.17	2.5	2.5
Protein, S. Cerev [42]	985	1.83	2.5	2.5
Software [43]	1376	6.39	2.5	2.5
Protein, S. cerev. [44]	1870	2.39	2.4	2.4
Internet, router [45]	3,888	2.57	2.48	2.48
Internet, domain [45]	4,389	3.76	2.2	2.2
Prot. Dom. (PromDom) [46]	5995	2.33	2.5	2.5

4.1 Performance Metrics

We select few suitable metrics as modularity index, NMI value to validate the performance measure of our proposed algorithm ACOGCT. Although there are some parametric measures as cost of clustering, cluster size of the algorithm to check the behaviour but these metrics provide important concepts of accuracy measurement. Graph size (number of nodes) is a basis, depending on which all the computation are executed to achieve the characteristics of the algorithm.

4.1.1 Modularity Index

A topology-based modularity metric, originally proposed by Newman and Girvan, 2004 [30], is used in this investigation to check the performance. This is a square symmetric matrix of clusters where each element d_{ij} represents the fraction of edges that link nodes between clusters i and j and each d_{ii} represents the fraction of edges linking nodes within cluster i. The modularity measure is given by Eq. (12) as follows.

$$M = \sum_i (d_{ii} - (\sum_j d_{ij})^2) \tag{12}$$

4.1.2 NMI Value

Another metric to estimate the quality of clusters achieved is the amount of mutual information shared between clusterings. This metric was originally defined by Kvalseth (1987) [47]. The NMI value plays an important role in checking the optimal nature of clusterings of different methods. It evaluates the algorithm's behaviour in information passing through different clustering results. It can predict the optimal or accurate clusters during clusterings. Assume, there are set of groupings of clusterings as $\{\lambda^{(q)} | q \in \{1,..,r\}\}$ which is denoted by $\wedge$. Let $n_h^{(a)}$ be the number of objects in the cluster c_h according to $\lambda^{(a)}$ and $n_l^{(b)}$ be the number of objects in the cluster c_l according to $\lambda^{(b)}$. Let $n_{h,l}$ represents the number of objects that are in c_h according to $\lambda^{(a)}$ and in cluster c_l according to $\lambda^{(b)}$.The symbol $\phi^{(NMI)}$ is denoted as the estimation of NMI (Kvalseth (1987)) as represented in Eq. (13).

$$\phi^{(NMI)}(\lambda^{(a)}\lambda^{(b)}) = \frac{\sum_{h=1}^{k^{(a)}}\sum_{l=1}^{k^{(b)}} n_{h,l}\log(\frac{n.n_{h,l}}{n_h^{(a)}n_l^{(b)}})}{\frac{(\sum_{l=1}^{k^{(b)}} n_l^{(b)}\log(\frac{n_l^{(b)}}{n})) + (\sum_{h=1}^{k^{(a)}} n_h^{(a)}\log(\frac{n_h^{(a)}}{n}))}{2}} \tag{13}$$

Based on this pairwise measure of mutual information, we can now define a measure between a *set* of *r* labelings, $\wedge$, and a *single* labelling $\lambda^{'}$ as the average normalized mutual information (ANMI) expressed by Eq. (14).

$$\phi^{(ANMI)}(\wedge, \lambda^{'}) = \frac{1}{r}\sum_{q=1}^{r}\phi^{(NMI)}(\lambda^{'}, \lambda^{(q)}) \tag{14}$$

4.1.3 Cluster size

Cluster size can determine the quality of clusters produced during clustering by any graph clustering algorithm. It is also computed as the number of clusters, produced from the clustering results.

4.2 Evaluation on Real-World Network Datasets

To obtain the performance measure on the basis of robustness and quality of the proposed algorithm ACOGCT compared to RNSC and MCL in the real-world scenario, few real power-law distribution datasets, placed in Table 1, are taken into account. The robustness and quality of the graph clustering algorithm are verified in terms of cost of clustering, cluster size, modularity index of clustering result and optimality.

4.2.1 Cost of Clustering: Table 2 gives the details of cost of clustering results, produced by ACOGCT, RNSC and MCL. The evaluation of cost is executed on real power-law distribution graphs with increasing graph size.

Table 2. Comparison of Cost of Clustering, Produced by ACOGCT, RNSC and MCL

Real Power-Law Networks	Cost of Clustering (ACOGCT)	Cost of Clustering (RNSC)	Cost of Clustering (MCL)
Electronic circuits	20326.06	20809.51	35552.36
Protein, S. Cerev	191633.2	197488.1	322559.9
Software	452441.6	452564.8	629417.7
Protein, S. cerev.	751914.3	781375.4	1163968
Internet, router	3131658.635	3233392	5032580
Internet, domain	4434223.79	4497642	6411685
Prot. Dom. (PromDom)	7552485.719	7738787	9652835

It is observed from Figure 4 that the cost is increased with increasing of graph size for all the test cases by these graph clustering algorithms. ACOGCT is performed better in cost evaluation compared to RNSC and MCL. The cost produced by MCL is more costly compared to ACOGCT and RNSC. But the cost produced by RNSC is less compared to MCL.

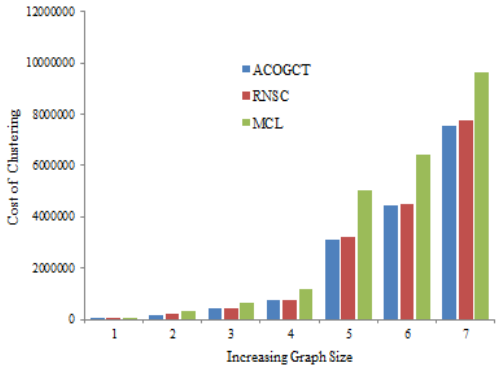

Figure 4. Evaluation of Cost of Clustering with Increasing Graph Size

It can be concluded that ACOGCT is producing lower-cost clustering results compared to RNSC and MCL.

4.2.2 Modularity Index: Table3 represents the modularity index values which are produced during clustering by ACOGCT, RNSC and MCL algorithm. The evaluation of modularity is performed on real power-law distribution graphs with increasing graph size.

Table 3. Comparison of Modularity Index of Clustering Results for ACOGCT, RNSC and MCL

Real Power-Law Networks	Modularity Index (ACOGCT)	Modularity Index (RNSC)	Modularity Index (MCL)
Electronic circuits	-12.17	-107.417	-14.6429
Protein, S. Cerev	-6.422	-354.502	-46.9323
Software	-3.946	-632.844	-57.5208
Protein, S. cerev.	79.9	-891.266	-117.11
Internet, router	-32.817	-1652.29	-243.728
Internet, domain	-36.557	-2343.96	-291.768
Prot. Dom. (PromDom)	-37.5893	-2523.57	-390.877

Modularity Index is an essential performance metric to test accuracy of clustering results of different graph clustering methods. The accuracy is measured based on the strength (intracluster links) of the clusters, produced during clustering. Figure 5 shows that ACOGCT is behaving more modular or producing more strength clusters compared to RNSC and MCL. The modularity is decreasing gradually with increasing of graph size in case of RNSC. MCL is achieving better modularity compared to RNSC. ACOGCT gains positive impact on modularity index evaluation. ACOGCT is producing more accurate clusters compared to RNSC and MCL.

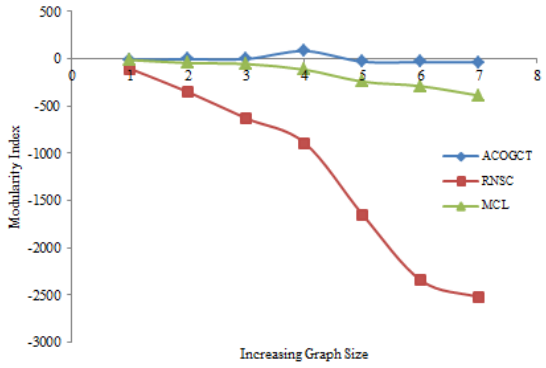

Figure 5. Modularity Index of Clustering Results with Increasing Graph Size

4.2.3 Cluster Size: Table 4 gives the detailed cluster size values, computed during clustering. The computation is performed on real power-law distribution graph with increasing graph size.

Table 4. Comparison of computed cluster size of graph clustering algorithms ACOGCT, RNSC and MCL

Real Power-Law Networks	Cluster Size (ACOGCT)	Cluster Size (RNSC)	Cluster Size (MCL)
Electronic circuits	119	187	54
Protein, S. Cerev	294	602	183
Software	418	805	397
Protein, S. cerev.	447	1311	253
Internet, router	1643	2589	700
Internet, domain	2212	3084	572
Prot. Dom. (PromDom)	2822	4003	1005

It is observed from figure 6 that cluster size prediction is nearly reaching the highest accuracy in case of ACOGCT compared to RNSC and MCL. This signifies that the rate of increment in cluster size with increasing of graph size is much better for ACOGCT. RNSC is producing huge number of clusters compared to ACOGCT and MCL. RNSC is not giving meaningful clusters. MCL is behaving not well in producing clusters.

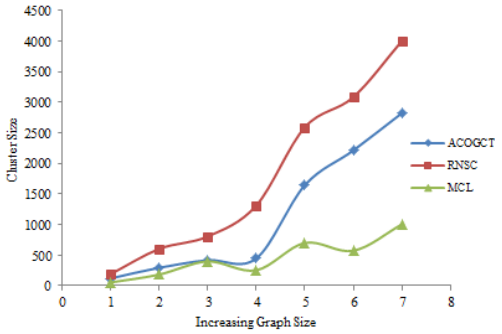

Figure 6. Evaluation of Cluster Size with Increasing Graph Size

The computed cluster size of MCL tells that MCL is not exploring the whole network properly. It can be concluded that ACOGCT is producing meaningful and significant clusters compared to RNSC and MCL.

4.3 Evaluation on Synthetic Dataset

The performance analysis is done on computer generated synthetic benchmark power-law distribution datasets. Power-law graphs are created by following the distribution principle which is mentioned before in eq (1). Basically the degree exponent γ is dispersed in the range of 2.1 to 3 for generating the scale-free model. In this paper, we use two different types of model of power-law distribution graphs. First model which is following the standard value of degree exponent as γ =2.5 and second model which is maintaining the degree exponent value ($\gamma = \frac{\ln 3}{\ln 2}$) of deterministic scale-free network. The deterministic scale free model is a hierarchical organization of hubs.

The performance measures are computed following the two model as two test cases.The results are shown for these two test cases as follows.

4.3.1 Cost of Clustering: It is observed from both Figure 7 and Figure 8 that the cost is increased with increasing of graph size. The cost of clustering is low always of ACOGCT for both the test case compared to RNSC and MCL. MCL is more costly that is shown in both the figure. RNSC is less costly compared to MCL for both the test cases. MCL is performing worst for the first test case. For the second test case, there is an acceptable balance of cost difference of the graph clustering algorithms whereas the first test case is not giving certain balance.

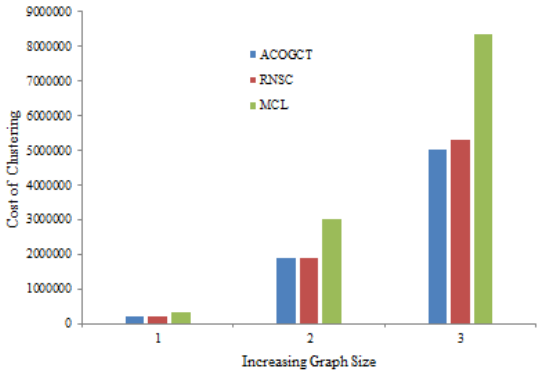

Figure 7. Cost of Clustering on Power-law Graph with Increasing Graph Size

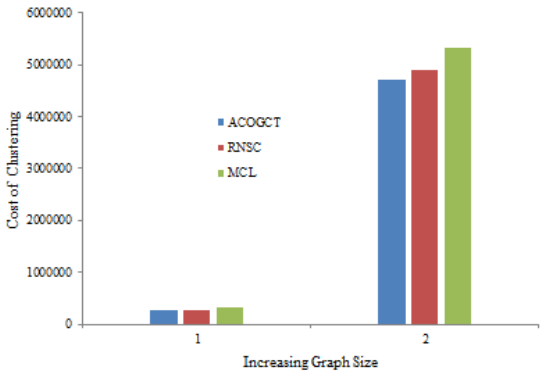

Figure 8. Cost of Clustering on Deterministic Power-law Graph with Increasing Graph Size

It can be concluded that ACOGCT is producing lower cost clustering for both the test cases compared to RNSC and MCL.

4.3.2 Modularity Index: Modularity is an important measurement technique which is used to justify the correctness of the clustering. It is observed from Figure 9 and Figure 10 that for both the two test cases ACOGCT is performing better compared to RNSC and MCL. RNSC is behaving more inferior compared to MCL for the two cases whereas MCL is performing well compared to RNSC.

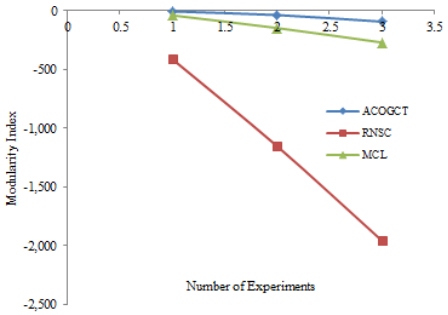

Figure 9. Modularity Index of Clustering Results on Power-law Graph

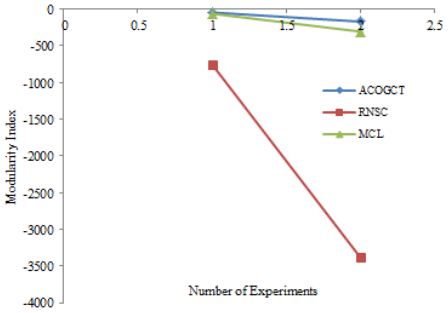

Figure 10. Modularity Index of Clustering Results on Deterministic Power-law

For the second test case, there is a high decrease in the modularity value for all the algorithms compared to the first test case. It can be determined that ACOGCT is more modular compared to RNSC and MCL for all the test cases.

4.3.3 Cluster Size: It can be easily understandable by detecting the cluster size that whether the clusters are good or bad. The cluster size estimation of both the Figures 11-12 shows that ACOGCT is producing good quality clusters compared to RNSC and MCL. RNSC is giving more number of clusters compared to ACOGCT and MCL. RNSC is not accurate in producing clusters. It can determine by seeing the MCL'S performance that MCL is not exploring the network properly for both the two test cases. For the second test case, cluster size curve for all the algorithms is not showing correct results compared to first test case.

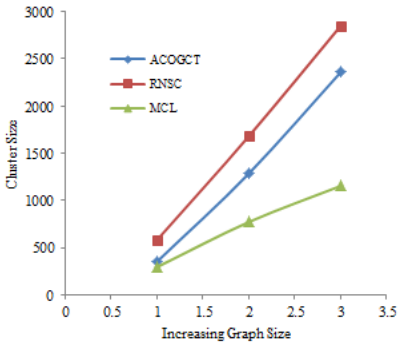

Figure 11. Cluster Size Estimation with Increasing Graph Size of Power-law Graphs

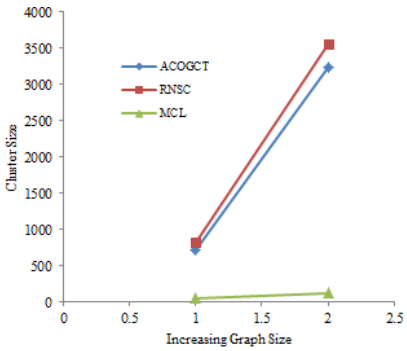

Figure 12. Cluster Size Estimation with Increasing Graph Size of Deterministic Power-law Graphs

There is a significant improvement in the cluster size prediction by ACOGCT compared to RNSC and MCL for all the test cases. ACOGCT is giving meaningful clusters compared to RNSC and MCL.

4.4 NMI Value on Real Power-law Distribution Graph

NMI is a significant approach which determines the quality of clusters of different graph clustering algorithms. The quality is measured in terms of optimality and it is basically obtained through the information passing between clustering results of a graph clustering algorithm. It is act as an information theoretic measure and shows the value of mutual information sharing between two clusterings of an algorithm.

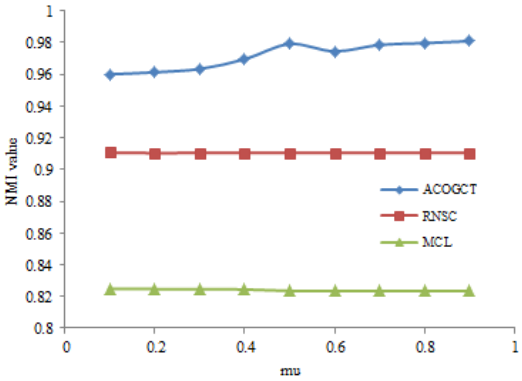

Figure 13. Estimation of NMI Value with mu

It is observed from Figure 13 that NMI value is highest in case of ACOGCT compared to RNSC and MCL. ACOGCT produces better quality clusters compared to RNSC and MCL whereas the clusters obtained from MCL clustering are not accurate. RNSC is giving optimal clusters compared to MCL. The Figure 13 is plotted using NMI value and mixing parameter (mu) of network with the range of 0.1 to 0.9. After 300, 500, 700 runs with using real large-scale power-law graph (Prot. Dom. [46]), NMI value is evaluated in case of ACOGCT and RNSC and in case of MCL; experiments are conducted by changing the inflation value as I= {2.5, 3.5, 4.5}. The NMI value curve shows that the optimality in increased with increasing of mu when mu>=0.4 the NMI is increasing gradually and it is very close to 1 in case of ACOGCT. RNSC and MCL are not showing that type of behaviour. It can be established that

the mutual information sharing between clusterings is more effective for ACOGCT whereas MCL can't provide good quality clusters due to the less NMI value. MCL is not giving accuracy in producing optimal clusters compared to RNSC also. It can be concluded that ACOGCT is producing meaningful clusters compared to RNSC and MCL. ACOGCT is more optimal compared to RNSC and MCL.

4.5 Visualization of Real and Synthetic Power-law Graph and Clustering

Figure 14 shows the visual representation of real power-law network software with 1376 nodes and huge interactions exist between the nodes following the power-law distribution. Figure 18 represents the visual presentation of deterministic power-law network with 1000 nodes. The visualization of clustering results, produced by ACOGCT, RNSC and MCL on real power-law graphs as software, internet, router, internet, domain and deterministic power-law network are represented in the following Figures 15, 17, 16. It can be resolved from the visualizations of clustering results that ACOGCT's clusters are more expressive and meaningful compared to RNSC and MCL. RNSC is performing better in producing clusters compared to MCL for all the test cases. It can be clearly assumed from the entire MCL's clustering that clusters are not properly visible and correct. RNSC is producing more optimal clusters compared to MCL. But the clusters, resulted from ACOGCT's clustering, are more accurate and proper compared to RNSC and MCL. ACOGCT is generating more optimal clusters compared to RNSC. ACOGCT is achieving significant improvement in producing optimal clusters for real large networks. All the visualizations of networks and clusterings are modularity controlled as they are shown in the following figures. Modularity is capable of identifying nodes which are in the same cluster using the help of some similarity measures, *i.e.*, basically determined using various properties of a complex network. It is perceived from fig 28 that clusters are marked appropriately by modularity approach in case of ACOGCT. RNSC and MCL are not responding well in that situation compared to ACOGCT. RNSC is behaving better compared to MCL in identification of clusters. The modularity approach is identifying the clusters, produced by ACOGCT accurately and the clusters are more expressive and significant compared to RNSC and MCL.

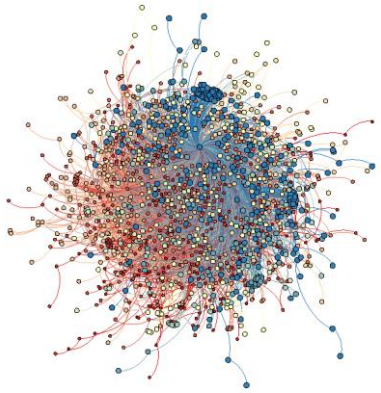

Figure 14. Visual representation of Software [43]

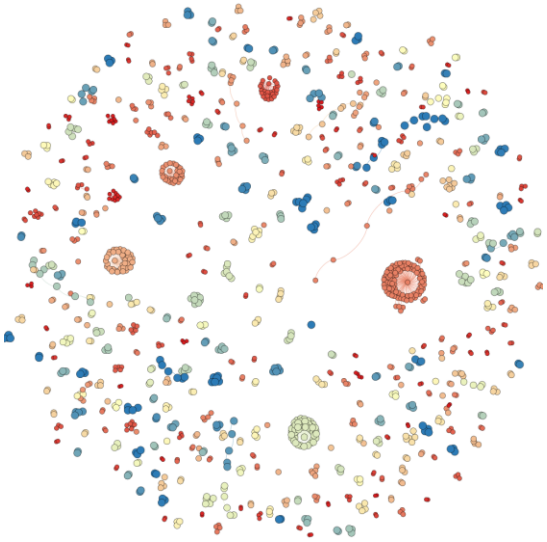

Figure 15. Visualization of ACOGCT's Clustering on Software

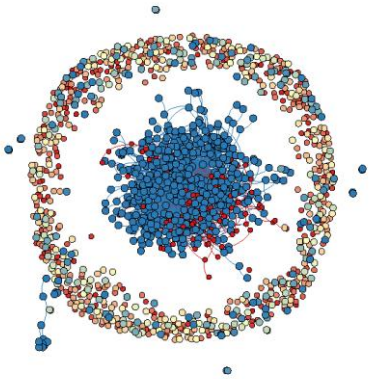

Figure 16. Visualization of MCL's Clustering on Software [43]

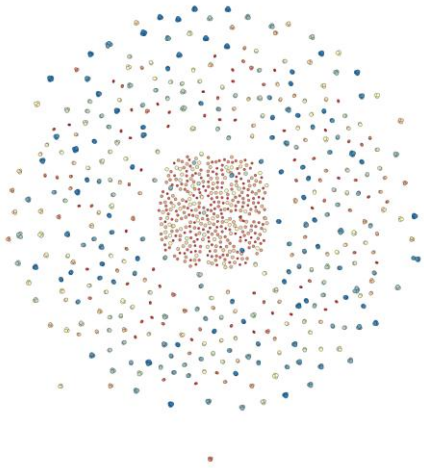

Figure 17. Visualization of RNSC's Clustering on Software [43]

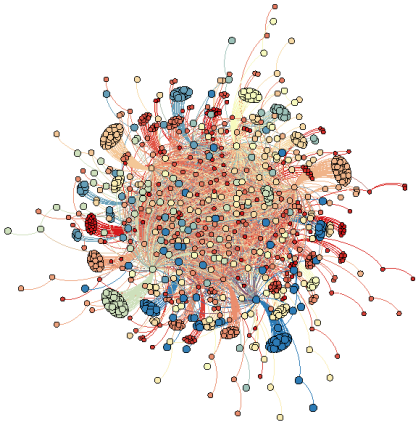

Figure 18. Visual representation of Deterministic Power-law Graph with 1000 Nodes

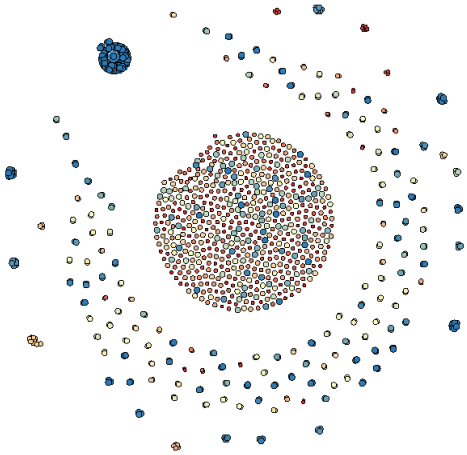

Figure 19. Visualization of ACOGCT's clustering on Deterministic Power-law Graph

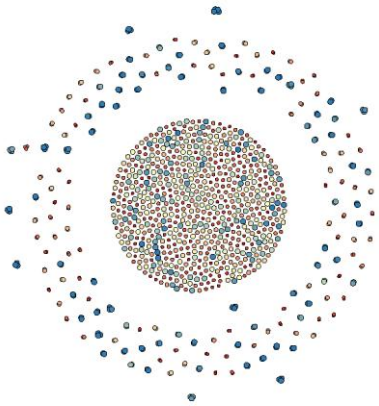

Figure 20. Visualization of RNSC's clustering on Deterministic Power-law Graph

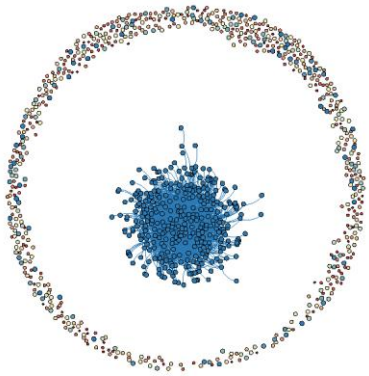

Figure 21. Visualization of MCL's clustering on Deterministic Power-law Graph

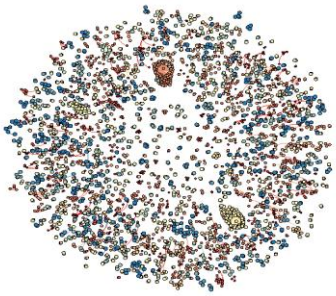

Figure 22. Visualization of ACOGCT's clustering on internet, router [45]

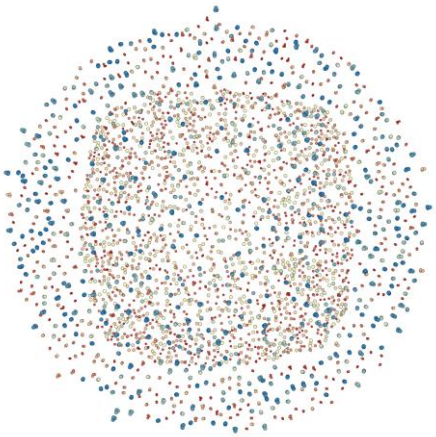

Figure 23. Visualization of RNSC's clustering on internet, router [45]

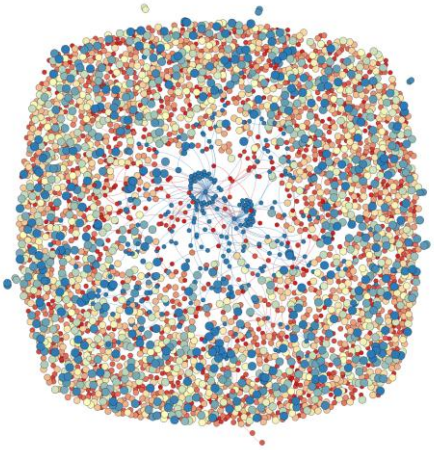

Figure 24. Visualization of MCL's Clustering on Internet, Router [45]

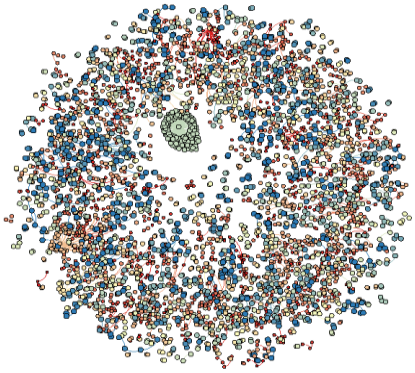

Figure 25. Visualization of ACOGCT's Clustering on Internet, Domain [45]

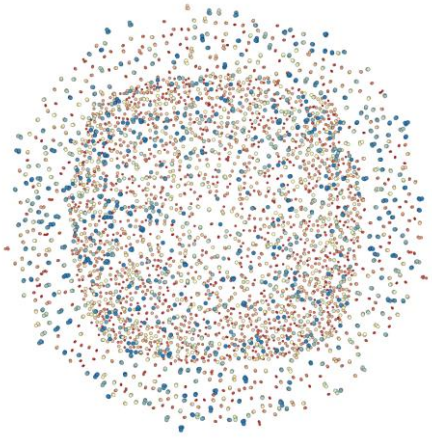

Figure 26. Visualization of RNSC's Clustering on Internet, Domain [45]

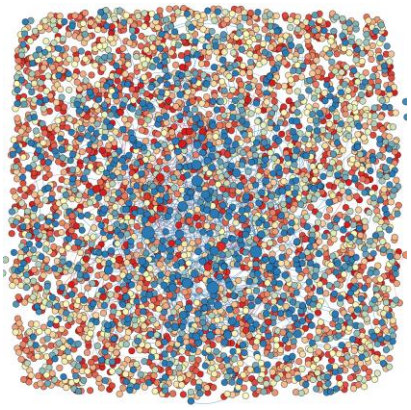

Figure 27. Visualization of MCL's Clustering on Internet, Domain [45]

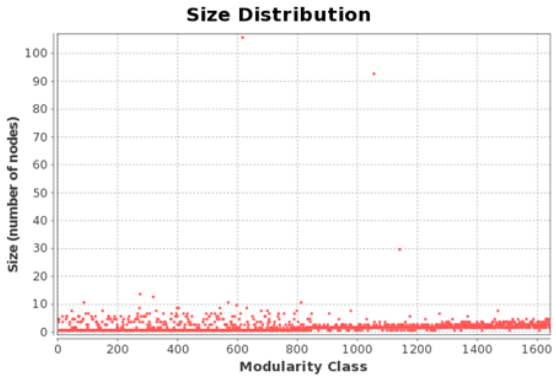

Figure 28. Modularity based Clusters Identifying in ACOGCT's Clustering on Internet, Router [45]

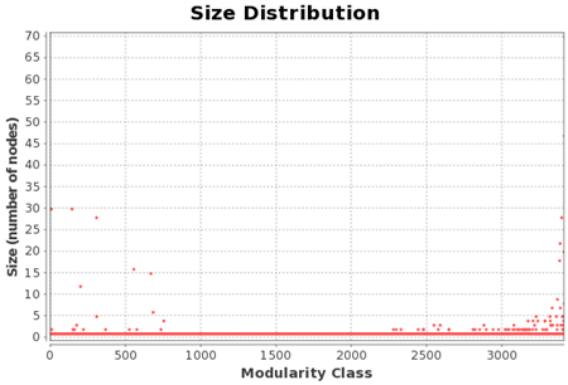

Figure 29. Modularity based clusters identifying in MCL's clustering on internet, router [45]

Figure 30. Modularity based clusters identifying in RNSC's clustering on internet, router [45]

5. Conclusions

Graph clustering is a fundamental task in many fields of science and engineering. Cost-efficient graph clustering algorithms are now-a-day a matter of investigation to satisfy some necessary aspects of real-world problems. In this paper, we present an aspiration criteria based graph clustering algorithm using the special effect of adaptive scaled cost function to detect high quality clusters in larger power-law networks. The scaled cost function, deduced by our algorithm ascertains the best (lower) cost clustering from large-scale networks compared with the baseline methods. The significance of an aspiration criterion based tabu search technique in our algorithm is to contribute an extra effort in achieving optimal cost clustering result from the set of clusterings of a network. Moreover it overcomes the shortcomings of optimality of clusters on the basis of NMI value that other graph clustering methods suffer from. Results on several synthetic and real benchmark power-law graphs highlight the utility of our approach when compared with RNSC and MCL on the basis of robustness and optimality. Scale cost evaluation is O (n) in RNSC. This can easily be done in O (1) time if the information about current node, and its cluster contribution are pre-computed and these features are incorporated in our algorithm. It can be further extended by a parallel move technique which will give better results in the case of run-time.

References

[1] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan-Kaufman, San Francisco, **(2006)**.
[2] A. Y. Ng, M. Jordan and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm", Proc. 14th Advances in Neural Information Processing Systems (NIPS '01), **(2001)**.
[3] B. A. Huberman, "Growth dynamics of the World-Wide Web", Nature, vol. 401, **(1999)**, pp. 131.
[4] M. E. J. Newman, S. H. Strogatz and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications", Physical Review E 64:026118, **(2001)**.
[5] W. Cai, S. Chen and D. Zhang, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation", Pattern Recognition, vol. 40, **(2007)**, pp. 825–838.
[6] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, **(1993)**, pp. 1101–1113.
[7] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data", Prentice-Hall, Englewood Cliffs, NJ **(1988)**.
[8] Z. Yu, H.S. Wong and H. Wang, "Graph-based consensus clustering for class discovery from gene expression data", Bioinformatics, vol. 23, **(2007)**, pp. 2888–2896.
[9] S. Bandyopadhyay, A. Mukhopadhyay and U. Maulik, "An improved algorithm for clustering gene expression data", Bioinformatics, vol. 23, **(2007)**, pp. 2859–2865.
[10] S. Wasserman and K. Faust, "Social Network Analysis: Methods and Applications", Cambridge University Press, **(1994)**.
[11] R. Guimera and L. Amaral, "Functional cartography of complex metabolic networks", Nature, vol. 433, no. 7028, **(2005)**, pp. 895–900.
[12] Y. Loewenstein, E. Portugaly, M. Fromer and M. Linial, "Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space", Bioinformatics, vol. 24, no. 13, **(2008)**, pp. i41–i49.
[13] Z. Chen, Y. He, P. Rosa-Neto, J. Germann and A. Evans, "Revealing modular architecture of human brain structural networks by using cortical thickness from MRI", Cerebral Cortex, vol. 18, no. 10, **(2008)**, pp. 2374–2381.
[14] W. Crum, "Spectral Clustering and label fusion for 3D tissue classification: Sensitivity and consistency analysis", in: Proceedings of Medical Image Understanding and Analysis, Dundee, Scotland **(2008)**.
[15] A. K. Jain, R. P. W. Duin and J. Mao, "Statistical pattern recognition: a review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, **(2000)**, pp. 4–37.
[16] A. Y. Ng, M. I. Jordan and Y. Weiss, "On spectral clustering: analysis and an algorithm", Advances in Neural Information Processing Systems vol. 14, MIT Press, Cambridge, MA, **(2002)**.

[17] A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: a review", ACM Computing Surveys, vol. 31, no. 3, **(1999)**, pp. 264–323.
[18] R. Xu and D. Wunsch II, "Survey of clustering algorithms", IEEE Transactions on Neural Networks, vol. 16, no. 3, **(2005)**, pp. 645–678.
[19] A. D. King, "Graph Clustering with Restricted Neighbourhood Search", M.S. Thesis, University of Toronto, **(2004)**.
[20] S. M. van Dongen, "Graph Clustering by Flow Simulation", PhD thesis, University of Utrecht, **(2002)**.
[21] F. Glover, "Tabu search", part I, ORSA Journal on Computing, vol. 1, no. 3, **(1989)** Summer, pp. 190-206.
[22] F. Glover, C. McMillan and B. Novick, "Interactive decision software and computer graphics for architectural and space planning", Annals of Operations Research, vol. 5, **(1985)**, pp. 557-573.
[23] F. Glover, "Tabu search", part II, ORSA Journal on Computing, vol. 2, no. 1, **(1990)** Winter, pp. 4-32.
[24] N. Mladenovi ć and P. Hansen, "Variable neighbourhood search", Computers and Operations Research, vol. 24, no. 11, **(1997)**, pp. 1097–1100.
[25] D. Harel and Y. Koren, "On clustering using random walks", In Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer S. Lecture Notes in Computer Science vol. 2245, Springer-Verlag, New York, **(2001)**, pp. 18–41.
[26] U. Brandes, M. Gaertler and D. Wagner, "Experiments on graph clustering algorithms", In Proceedings of the 11th Annual European Symposium on Algorithms (ESA'03), Lecture Notes in Computer Science vol. 2832, **(2003)**, pp. 568–579.
[27] S. Vempala, R. Kannan and A. Vetta, "On clusterings—good, bad and spectral", In Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS'00), **(2000)**, pp. 367–378.
[28] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity", Information Processing Letters, vol. 76, no. 4-6, **(2000)**, pp. 175–181**.**
[29] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity", Information Processing Letters, vol. 76, no. 4-6, (**2000)**, pp. 175-181.
[30] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks", Physical Review E 69, 026113, **(2004).**
[31] A. Clauset, M. E. J. Newman and C. Moore, "Finding community structure in very large networks", Physical Review E 70, 066111, **(2004**).
[32] H. A. D. do Nascimento and P. Eades, "A system for graph clustering based on user hints", In Peter Eades and Jesse Jin, (Eds.), Selected papers from Pan-Sydney Workshop on Visual Information Processing, Sydney, Australia, ACS (**2001)**.
[33] H. H. Hoos and T. Stutzle, "Systematic vs. local search for SAT", InWolfram Burgard, Thomas Christaller, and Armin B. Cremers, editors, KI-99: Advances in Artificial Intelligence, 23rd Annual German Conference on Artificial Intelligence, Bonn, Germany, (**1999)** September, pp. 298-293.
[34] R. Albert and A.-L. Barabasi, Rev. Mod. Phys. in press cond-mat/0106144, (**2001**).
[35] L. A. N. Amaral, A. Scala, M. Barthelemy and H. E. Stanley, "Classes of small-world networks", Proc. Natl. Acad. Sci. USA, vol. 97, no. 11, (**2000**), pp. 149.
[36] M. Faloutsos, P. Faloutsos and C. Faloutsos, "Weighted Scale-free Networks in Euclidean Space Using Local Selection Rule", Proceedings of the ACM SIGCOMM, Comput. Commun. Rev., vol. 29, (**1999**), pp. 251.
[37] R. Albert, H. Jeong and A. -L. Barabasi, "Diameter of the World-Wide Web", Nature, vol. 401, (**1999**), pp. 130.
[38] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai and A. -L. Barabasi, "The large-scale organization of metabolic networks", Nature, vol. 407, (**2000**), pp. 651–654.
[39] A. -L. Barabasi and R. Albert, "Emergence of Scaling in Random Networks", Science, vol. 286, (**1998**), pp. 509.
[40] C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields and P. Bork, "Comparative assessment of large-scale data sets of protein-protein interactions", Nature, vol. 417, no. 6887, (**2002)**, pp. 399-403.
[41] R. F. i. Cancho, C. Janssen and R. V. Sole, "Topology of technology graphs: small world patterns in electronic circuits", Phys. Rev. E, vol. 64, 046119, (**2001**).
[42] A. Wagner, "the Yeast Protein Interaction Network Evolves Rapidly and Contains Few Redundant Duplicate Genes", Mol. Biol. Evol., vol. 18, (**2001**), pp. 1283-1292.
[43] S. Valverde, R. Ferrer-Cancho and R. V. Sole, "Scale-Free Networks from optimal design", arXiv: cond-mat/0204344, **(2002)**.
[44] H. Jeong, S. Mason, A. -L. Barab ási and Z. N. Oltvai, "Lethality and centrality in protein networks", Nature, vol. 411, (**2001**), pp. 41-42.
[45] M. Faloutsos, P. Faloutsos and C. Faloutsos, "on power-law relationships of the Internet topology", Comput. Commun. Rev., vol. 29, (**1999**), pp. 251-262.
[46] S. Wuchty, "Scale-Free Behavior in Protein Domain Networks", Mol. Biol. Evol., vol. 18, (**2001**), pp. 1699-1702.

[47] T. O. Kvalseth, "Entropy and correlation: Some comments", Systems, Man and Cybernetics, IEEE Transactions, vol. 17, no. 3, **(1987)**, pp. 517–519**.**
[48] M. Dhara and K. K.Shukla, "Comparative performance analysis of RNSC and MCL algorithms on power-law distribution", Advanced Computing: An International Journal (ACIJ), vol. 3, no. 5, (**2012**) September, pp. 19-34.
[49] M. Dhara and K. K. Shukla, "Performance Testing of RNSC and MCL Algorithms on Random Geometric Graphs", International Journal of Computer Applications (0975 – 8887), vol. 53, no. 12, (**2012**) September.

Authors

Mousumi Dhara. She completed her M. Tech in computer Science and Engineering from NIT Durgapur. She is pursuing her Ph.D. since 2010 in the Department of Computer Engineering, IIT (BHU) and Varanasi.

K. K. Shukla. He is professor of Department of Computer Engineering, Indian Institute of Technology, Banaras Hindu University and Varanasi.