International Journal of Engineering & Technology, 5 (x) (2017) xxx-xxx

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

doi: xxx

Research paper, Short communication, Review, Technical paper

Graph Coloring With Social Media Networking Using Hadoop

Neenu Jose[1*] and Joseph John[2]

[1]*PG Student, Dept. of Computer Science, Kalam Technological University, RSET*
[2]*Assistant Professor, Dept. of Computer Science, Kalam Technological University, RSET*
[*]*E-mail: neenujose20@gmail.com, josephj@rajagiritech.edu.in*

Abstract

In graph coloring problem (GCP), a graph has been colored such that a color is assigned to each vertex in a way, adjacent vertices will have different colors. In modern sociology, Social network evaluation has emerged as a key technique. Graphs used to illustrate relationships between humans.To store such large data certain data analytic frameworks are available like hadoop. Inorder to solve real world problems, these data are often pictured in large scale graphs.The main aim is to identify the key nodes of communication in social media networking using In-Betweenness and graph coloring. Local Smallest-Largest Degree First[1] is the algorithm considered for graph coloring.The algorithm executed in each superstep and the vertex value, which indicate whether vertex is colored or not will be updated in each superstep. In-Betweenness is the measure for analysing the control of a human on the verbal exchange between different people in a social network. The measure of centrality in a graph is known as betweenness centrality which is based on shortest paths. The number of shortest paths that pass through the vertex is considered as the centrality value of that vertex. Key nodes are identified using Inbetweenness and that nodes will have more oversight over the network such that more information will pass through that nodes. Centrality value of a vertex can be calculated by using Brande's algorithm. Breadth First Search algorithm is used for calculating shortest path in Brandes algorithm. Based on the key nodes, graph coloring is performed and it will be colored uniquely if there is connection between those key nodes. Both constraints like degree and betweenness centrality are considered for coloring. Results and analysis shows the comparison on implementing the concept of inbetweenness and without inbetweenness based on execution time, number of super steps and number of colors in both single and multi node hadoop clustering. Therefore in this paper we propose a hadoop framework for graph coloring using social media and we have found that results are promising. It gives an insight that, with and without inbetweenness there is an execution time difference which can be utilized for further realtime applications.

Keywords: *Betweenness Centrality;Graph;Graph Coloring;Local Smallest-Largest Degree First;Social Media;Superstep*

1. Introduction

Graph coloring is an anomaly of graph labelling. Graph coloring includes vertex coloring, face coloring and edge coloring. Vertex coloring in simple words is the assigning of colors to vertices where no two connected vertices have the same color.Capability of a technology can be accessed using these constraints. The least number of colors used for coloring of graph is called chromatic number.A graph is k-colorable, when it can be allot a (proper) k-coloring and if its chromatic number is just k,then it is k-chromatic.Suppose if we want to group persons in n commission of k individuals, such that to give their point of view on some topics. Based on the importance, the topics are assigned to more than one commission. Care should be taken that they don't have more than i persons in common for each commission.Let G=(V,E) be the graph used to model this problem, in which each vertex conforms to a commission and if the corresponding commissions share at least one topic, there is an edge between a pair of vertices.There are different types of graphs like simple graph, null graph, complete graph,planar graph each of having different characteristics. If two vertices share an edge then they are connected to each other.Vertices which comes in boundary will refers to boundary vertices and vertices which are not boundary is known as interior vertices.Path which is not having vertices in boundary is known as internal path. Colors will be assigned to each vertex where colors refers to persons. The problem is to find the minimal number of colors used to assign each vertex such that neighboring vertices don't share same color[10]. some times numbers 1,2..etc refers to colors which is used to color the graph. Graph coloring can also be performed in digital form.

In other words,Graph coloring is only a simple method for labelling graph, for example, vertices, edges, and locales under a few limitations. In graph coloring, no two nearby vertices, adjoining edges, or neighbouring locales are colored with same color and with minimum color. This number is known as the chromatic number and the graph is known as a properly colored graph. In various research areas like data mining, clustering, image capturing and segmentation, networking..etc graph coloring is used. The most vital background of graph coloring is also used in resource allocation and scheduling.

Social community analysis has emerged as a key strategy in current sociology [12]. Figure 1 shows the graphical representation of a social media network. In the graph, vertices indicate people and if there is an edge between two vertices,it indicates communication link between them. In that figure graph is just labelled based on the number of communication link. Blue colored area indicates the area which is having more communication links. Social network

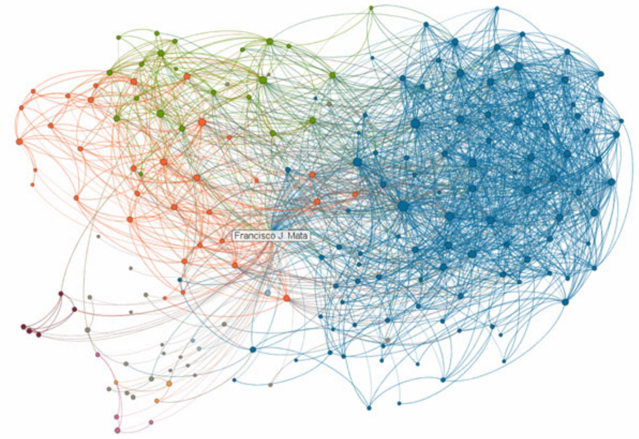

Figure 1: Graphical representation of a social media network[13]. In the graph, vertices indicate people and if there is an edge between two vertices,it indicates communication link between them.

analysis characterizes networked constructions in phrases of nodes (individual actors or people) and the ties, edges, or hyperlinks (relationships or interactions) that join them.The social layout in the Internet context is a graph that depicts non-public household individuals of web users. Betweenness centrality is a measure of centrality in a graph which is based on shortest paths.Figure 2 represents the concept of betweenness centality.Red color indicates those which have lesser betweenness centrality value and blue color indicates the one which is having higher centrality value. For each and every pair of vertices in a connected graph, there will be at least one shortest route between the vertices such that both the variety of edges that the direction passes via (for unweighted graphs) or the summation of the weights of those edges (for weighted graphs) must minimized. The betweenness centrality for every vertex is the range of these shortest paths that passes via the vertex. It finds wide utility in network theory and represents the degree of which nodes stand between other vertices. More data will moves through the node which is having large centrality value. Therefore that node will have extra influence over the network. Therefore the problem statement is that, identify the key nodes of communication in social media networking using the concept of In-Betweenness and graph coloring.

In this paper we use hadoop framework. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models or in other words, Hadoop is an open-source structure that permits to store and process huge information in an appropriated domain crosswise over groups of PCs utilizing basic programming models.It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.Big data implies as an accumulation of expansive datasets that can't be handled using regular databases. Big data isn't just an information, rather it has turned into a total subject, which includes different apparatuses, techniques and systems.Big data innovations are essential in giving more precise investigation, which may prompt more solid basic leadership bringing about more noteworthy operational efficiency's, cost decreases, and lessened dangers for the business. To bridle the energy of big data, you would require a framework that can oversee and process colossal volumes of organized and unstructured information in real time and can ensure information protection and security.

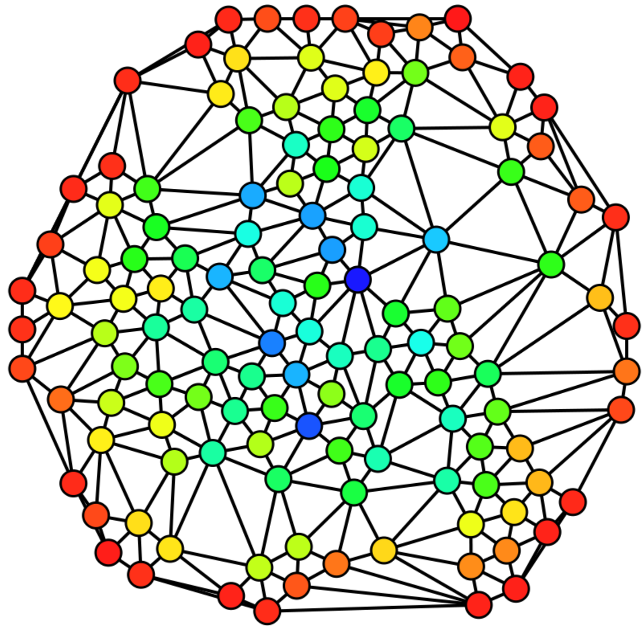

Figure 2: Graphical representation of Inbetweenness[14]. Red color indicates those which have lesser betweenness centrality value and blue color indicates the one which is having higher centrality value.

2. Background Study

There are different techniques available for graph coloring like greedy, neural network, watermarking technique, ant colony, genetic algorithm, particle swarm optimization etc. Some of those techniques are discussed below as a part of background study.

Greedy algorithm is an algorithm model that follows the method in which chooses an optimal solution in each stage.This is the very basic and simple technique used for graph coloring and it will give optimal solution. There are different greedy algorithms available for graph coloring and the guarantee which is given by the basic algorithm is that they won't use any more number of colors than the number of vertices. Andrew Lim and Fan Wang[2] suggests an Extension of classical algorithm like meta heuristic. This method focuses on building robust coloring using fixed number of colors.It also considers the possibility of correcting those coloring where both vertices of an missing edge share the same color Partition based encoding is used in search space. Different sizes of input are required.In [2] we can see that depending up on the input size we can choose different meta heuristic available. The execution time and complexity depends up on the heuristic. But the problem arise in case of big data.

Neural network is type of artificial intelligence which try to follow the way in which a brain works. Q'tron neural network model is used for solving graph coloring problem and it is known as energy system is proposed by Tai-Wen Yue,Zou Zhong Lee[3]. By making free the local minima they attained a goal directed search.The model is developed by placing all the feasible coloring scheme in global minima. Suppose K colors are used for solving the problem, then by decreasing the values of k goal can be achieved. The network is settle down only if the goal is reachable. The proposed models experimental results is compared with DSATUR algorithm. Becomes complex in case of bigdata. Ant colony is a meta heuristic method in which, for a given optimization problem, artificial ants as a set of software agents search for a good solution.The method iteratively set up the solution in construction strategy is used in [4]. Tabu search is used in improvement strategy. Constructive methods used are RLF and DSATUR. Tabu search moves from a coloring to another. Based on the used constructive method, heuristic information is defined. But this needs more execution time in case of execution time.

The process of hiding digital information is known as watermarking.The problem of graph coloring is considered online by Sundar Vishwanathan [5]. One vertex of the graph is presented at a time. As soon as the vertex is presented, online algorithm should assign a color to a vertex. The objective of an online algorithm is to color the graph efficiently. In order to mend on the color of each vertex, the algorithm takes polynomial amount of time. Graph and the arrangement of the vertices is determined by the assailant . It is fixed for the rest and presents at a time one vertex. The residue set is the set which contain the remaining vertices which is not colored.In residue set, each vertex will be adjacent to some of the vertices in greedy class. Before moving to next vertex, color the current vertex within exponential amount of time. Smallest admissible color is assigned to each vertex by greedy algorithm. This technique doesn't work well in general graphs.

Evolutionary ideas like natural selection and genetics is followed in genetic algorithms. Aim is to obtain a minimal chromatic number and it is achieved by using genetic divide and conquer method is used in [6]. The vertex set id divide in to different subsets and then solved. The new algorithm is proposed using Single Parent Conflict Gene crossover (SPCGX) conflict edge mutation operators. Stochastic convergence is also analyzed based on certain constraints. Experimental results show that, the method is work well even in minimal population. This devised method reduces complexity as compared with the basic genetic algorithm.Complexity and big data handling is tough in some cases of genetic algorithms.In Particle swarm optimization the particles which is known as potential solutions will fly through the problem space.But it wont have any crossover or mutation operators as there in genetic algorithm. akuya Aoki, Claus Aranha, Hitoshi Kanoh[9] suggests a solution based on hamming distance. Particle Swarm Optimization (PSO) is a population based search algorithm. In the search space number of individuals named particles are placed. In this graph 3 coloring problem is considered.Graph 3 coloring problem is that 3 colors are used for coloring. Red, Green and Blue are the colors they considered. Hamming distance is that which is used to find the distance between particles. Transition probabilities is used to find the internal relationship between the particles. Operation of transition probability is changed at last and a new transition probability is formulated.

In distributed systems, messages will be passed through network in-order to have communication between computers. The problem will be separated in to different tasks and each system will be assigned different tasks. Algorithms for Balanced Graph Colorings with Applications in Parallel Computing[8] mainly aims on attaining balance coloring in parallel computing. Distance coloring and partial balance 2 coloring are the two variants considered. Three different guided balance algorithms are proposed and parallelism those algorithms. ab initio approach is used for balance coloring and the other used guided approach. Guided approach used scheduled and unscheduled moves. Inorder to manipulate large amount of data Hadoop is used in [1]. Graph coloring is incorporated in to hadoop. In this paper they have compared four parallel algorithms. Bulk synchronous parallel model is used over hadoop and it consist of supersteps. Local Maxima First, Local Minima-Maxima First, Local Largest Degree First, Local Smallest-Largest Degree First are the algorithms implemented. Experimental results give that different algorithm is better in different constraints.

Since we are handling a social media networking dataset, on comparing with other techniques available for graph coloring distributed systems will be more convenient in case of execution time and complexity. In [1] social dataset is handled using hadoop and among those algorithms, local smallest-largest degree first algorithm is more efficient since it is performed well than other algorithms. So in the proposed technique this algorithm is chosen by incorporating the concept of inbetweenness.

3. Problem Statement

Social network analysis has developed as a key methodology in current humanism. It portrays arranged developments in expressions of hubs (singular on-screen characters or individuals) and the ties, edges, or hyperlinks (connections or collaborations) that go along with them. Researching social structures is using systems and organization hypothesis. Graphs used to speak to connections between people. In-Betweenness is the Measure for evaluating the control of a human on the correspondence between other individuals in an informal organization. Through that we can identify the person which is having more control over the network.

Therefore the problem statement is, to identify the key nodes of communication in social media networking using In-Betweenness and graph coloring.

4. Proposed Method

The aim is to identify the key nodes of communication in social media networking using In-Betweenness and graph coloring. Local Smallest-Largest Degree First[1] is considered for graph coloring and inbetweenness is included in to the algorithm and comparison is made based on some criteria. The proposed algorithm is implemented in hadoop. Program consist of a sequence of Supersteps (S0,S1,S2,...,Sn). The input to the program is a Graph (G0) in the form of set of tuples. Certain presumptions are considered in this algorithm. These algorithms are for undirected graph. Each vertex is assumed to be associated with special number. This special quantity of vertex is certainly its VertexId. Each vertex additionally has one variable to store any form of data, which is known as VertexValue. At the give up of computation, the assigned coloration for each vertex is saved in VertexValue. Basically contain 4 modules and the modules are explained below.

4.1. Preprocessing

This is the initial module. In preprocessing module, the input will be the data set. Data set is social media dataset which contain set of vertices. The dataset that we have used here is facebook dataset. The input will be splitted up in such a format like vertex and its corresponding neighboring vertices. This will be the input to the next module.

4.2. Message Setting

The message which has to send to each neighboring vertices is calculated in this module. The input to the module is set of vertices and its neighboring ones. Initially in this module the vertex value is assigned. The message which has to send to each neighboring vertex is calculated. The message contain two parts, contains vertex id and the number of edges of that particular vertex. This will be send to each of the connected vertices. For example, suppose the degree of a vertex is n, then that particular edge will get n number of messages. This whole module is represented as a map-reduce module. The output from this map-reduce module is a key value pair where the key is the vertexid and the value is the message which contain two parts.

4.3. Inbetweenness

Betweenness centrality for a vertex v is described in phrases of the share of shortest paths that passes through v [11]. Brandes algorithm is used for computing the betweenness centrality.Initially we have to set the centrality value of each vertex as zero. The shortest path between two nodes is calculated by using breadth first search (BFS) algorithm and it will be stored.In Brandes algorithm the centrality value is calculated as dividing the number of shortest paths between two nodes which passes through the node which is being considered by the total number of shortest paths between two nodes.

key node is identified from this module. The node which is having higher value for betweenness centrality will have higher influence on that particular network. It will control the whole network. This module is actually represented as a map-reduce module where the input to the maper function is the key,value pair from the previous module. Based on that key,value pair the key node is identified by implementing the brande's algorithm.

4.4. Graph Coloring

The colored graph is obtained from this module. The input to this module is key node. Based on the key node graph coloring algorithm is implemented. As already mentioned the algorithm choosen for graph coloring is Local Smallest Largest Degree First.

The colored graph is obtained from this module.The algorithm is given in Algorithm 1. The input to this module is key nodes. Based on the key nodes graph coloring algorithm is implemented. As already mentioned the algorithm chosen for graph coloring is Local Smallest Largest Degree First[1]. In the algorithm Vertex.value indicates the color which is being assigned to the vertex,that value will be updated in each superstep. Initially it will be assigned as -1. Message contain three parts Vertex.id which represents the unique number which is given to each of the vertex, Vertex.getNumEdges() indicates the number of edges of that vertex and Vertex.getBetweenness() will give the centrality value of each vertex. We will get each part in the message by using function contain(). Create two independent set at a time using degree of vertex and color them with two different colors.For uncolored vertices, we pick out set of vertices with smallest degree and set of vertices with biggest degree at a same time. In the battle of same vertex degree of neighbors, the VertexId is used to wreck the tie. In case of tie, vertex with larger VertexId is in addition regarded for set of greatest degree vertices and vertex with smallest VertexId is viewed for set of smallest degree vertices in present superstep.The method of this algorithm reduces the wide variety of supersteps and gives higher runtime performance than the unique algorithms used for format coloring. Colored graph will be obtained after the execution of the algorithm indicating the key node which is colored initially.

Result: Colored graph in which Key nodes are uniquely labelled
Void Compute(Vertex(Long,Double,Float), Message,(Double));
Set MaxVal=true;
Set MinVal=true;
if *Superstep=0* **then**
  Set Vertex.Value=-1;
  Message=Vertex.id + Vertex.getNumEdge() + Vertex.getBetweenness();
  Send Message to all the Connected Vertices;
else
  if *Vertex.Value=-1* **then**
    while *each Message: msg* **do**
      VertexIdMsg=msg.contain(1);
      NumEdgeMsg=msg.contain(2);
      BetweennessMsg=msg.contain(3);
      if *Vertex.getNumEdge() < NumEdgeMsg and Vertex.getBetweenness < BetweennessMsg* **then**
        MaxVal=false;
      end
      else if *Vertex.getNumEdge () > NumEdgeMsg and Vertex.getBetweenness > BetweennessMsg* **then**
      MinVal=false ;
      else if *Vertex.getNumEdge() = NumEdgeMsg and Vertex.getBetweenness = BetweennessMsg* **then**
      if *VertexIdMsg > Vertex.Id* **then**
        MaxVal=false;
      end
      if *VertexIdMsg < Vertex.Id* **then**
        MinVal=false ;
      end
    end
    if *MaxVal = true* **then**
      Set Vertex.Value=$2 \times CurrentSuperstep$ **end**
      else if *MinVal = true* **then**
      SetVertex.Value=$2 \times CurrentSuperstep - 1$
      Message=Vertex.id+Vertex.getNumEdge()+ Vertex.getBetweenness();
      Send Message to all the Neighbors;
    end
  end
  VoteToHalt();

Algorithm 1: Algorithm For Graph Coloring With Betweenness Centrality

5. Results and Analysis

This section discusses about the results obtained after the implementation of proposed algorithm. As a part of experimental setup, we have built hadoop cluster with two nodes. We have considered both 16GB and 8GB RAM for each node in hadoop cluster. Operating system used for each node is Ubuntu desktop 16.04 and hadoop version is 2.6.5 subversion. The dataset we have used is facebook dataset which is available for research [15].The method is implemented in both single node and multi node hadoop cluster.

Figure 3 shows the output obtained after implementing message setting module. The betweenness centality value is calculated and it will send to each of the connected vertices. In that we can see that the message which is recieved by each of the vertex contain the id, degree and centrality value of each of the connected edges. That will be considered for graph coloring. The vertex value indicate the value of color.

```
INFO:
***********************        1       476.43642330103216 ********
Message{vID=299, vDegree=38, bCentrality=81.33617544995552}
Message{vID=194, vDegree=36, bCentrality=29.321472017771153}
Message{vID=346, vDegree=52, bCentrality=511.3510615235305}
Message{vID=126, vDegree=12, bCentrality=17.961374733143366}
Message{vID=315, vDegree=110, bCentrality=3564.0297635418565}
Message{vID=48, vDegree=42, bCentrality=838.7491505255541}
Message{vID=88, vDegree=38, bCentrality=590.3405049667592}
Message{vID=236, vDegree=72, bCentrality=400.3012143829716}
Message{vID=54, vDegree=14, bCentrality=47.74076309971763}
Message{vID=322, vDegree=142, bCentrality=4742.830427325284}
Message{vID=280, vDegree=84, bCentrality=1462.940043758468}
Message{vID=92, vDegree=40, bCentrality=41.97433539188884}
Message{vID=73, vDegree=18, bCentrality=115.44660822511119}
Message{vID=133, vDegree=34, bCentrality=718.8991229478254}
Message{vID=119, vDegree=122, bCentrality=8498.357476461762}
Message{vID=53, vDegree=60, bCentrality=1231.7666592358357}
****************************************
***********************        2       19.43718878492146 *********
Message{vID=226, vDegree=26, bCentrality=43.313211228656705}
Message{vID=312, vDegree=50, bCentrality=2876.899903505802}
Message{vID=333, vDegree=14, bCentrality=6.152846424718754}
Message{vID=343, vDegree=34, bCentrality=942.2739103564087}
Message{vID=20, vDegree=28, bCentrality=40.61796366255971}
Message{vID=149, vDegree=26, bCentrality=42.76374735038923}
Message{vID=326, vDegree=36, bCentrality=107.01960062611941}
Message{vID=115, vDegree=40, bCentrality=4960.227143707612}
Message{vID=116, vDegree=32, bCentrality=3082.371424627436}
****************************************
***********************        3       564.5415817731443 *********
Message{vID=200, vDegree=112, bCentrality=815.0729685520222}
Message{vID=280, vDegree=84, bCentrality=1462.940043758468}
```

Figure 3: Output obtained after the execution of message setting module. It contain vertex id, message received and the centrality value of that vertex. The message contain the vertexid and centrality of the connected vertices.

5.1. Comparison

Comparison is done based on execution time, dataset and RAM. The algorithm is compared with Local Smallest-Largest Degree algorithm. Table 1 show the results got when we implemented the method in 8GB RAM by using the dataset which contain 5k edges. We can see that even if the execution time is increased by using betweenness, not only the number of superstep but also the number of colors used are decreased. Table 2 shows the result obtained when the method is implemented in 16GB RAM by using same dataset as in 8GB RAM. We can see that the execution time is decreased than found in 8GB RAM. Table 3 shows the results obtained by using the dataset which contain 1Lakh edges. On comparing the results of Table 1 and Table 2 we can see that as the core increases execution time decreases and Large amount of data can be executed more easily as in Table 3. Results obtained after implementing the method in single node and multi node clustering is also used for comparison. Figure 4 shows the comparison based on number of supersteps which is implemented in single node cluster based on betweenness centrality. From the graph we can infer that, the number of supersteps get reduced by using betweenness centrality. Figure 5 shows the comparison based on number of colors which is implemented in single node cluster based on betweenness centrality. From the graph we can infer that, the number of colors get reduced and become more optimized by using betweenness centrality. The method is also implemented in multi node cluster. Figure 6 shows the comparison of results obtained when graph coloring algorithm with betweenness centrality is implemented, in both single node and multi node clustering. From the graph we can see that as the number of computational nodes is inversely proportional to execution time.

6. Conclusion and Future Work

In graph coloring problem (GCP), a graph has been colored such that a color is assigned to each vertex in a way that adjacent vertices will have different colors. In modern sociology, Social network evaluation has emerged as a key technique. Graphs used to illustrate relationships between humans.In- Betweenness is the measure for analysing the control of a human on the verbal

Table 1: Results obtained in 8GB RAM (5k edges)

	No: of Vertices	No: of Super steps	No: of Colors Used For Coloring	Execution Time (minutes)
Graph Coloring Without Inbetweenness	333 (5k edges)	25	50	3779
Graph Coloring Without Inbetweenness	333 (5k edges)	24	48	4865

Table 2: Results obtained in 16GB RAM(5k Edges)

	No: of Vertices	No: of Super steps	No: of Colors Used For Coloring	Execution Time (minutes)
Graph Coloring Without Inbetweenness	333 (5k edges)	25	50	2905
Graph Coloring Without Inbetweenness	333 (5k edges)	24	48	3409

Table 3: Results obtained in 16GB RAM(1Lakh Edges)

	No: of Vertices	No: of Super steps	No: of Colors Used For Coloring	Execution Time (minutes)
Graph Coloring Without Inbetweenness	4039 (1 Lakh edges)	131	262	8
Graph Coloring Without Inbetweenness	4039 (1 Lakh edges)	120	239	12

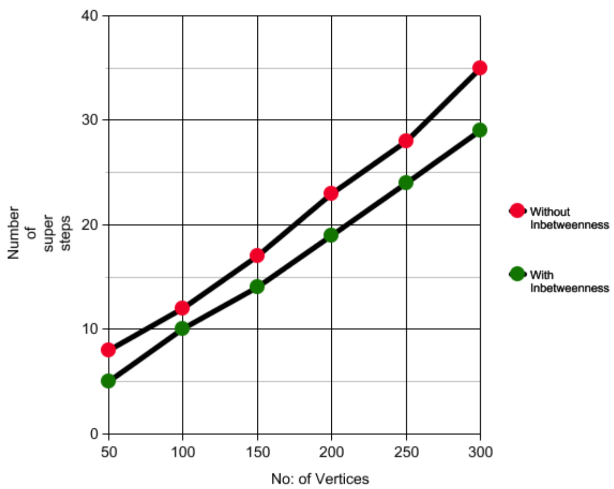

Figure 4: Comparison of results based on number of supersteps when both concepts are implemented in single node hadoop cluster.

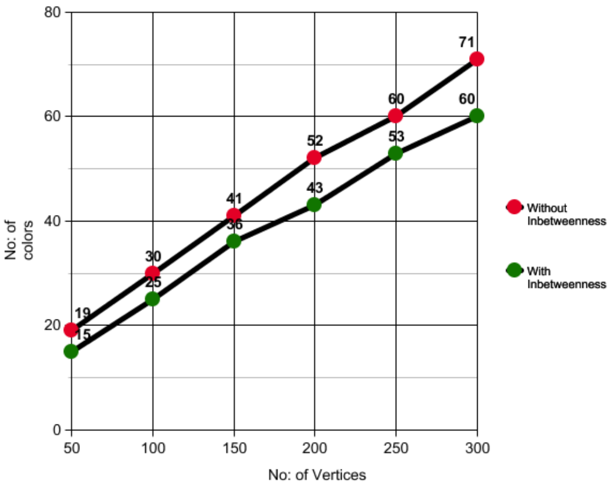

Figure 5: Comparison of results based on number of colors when both concepts are implemented in single node hadoop cluster.

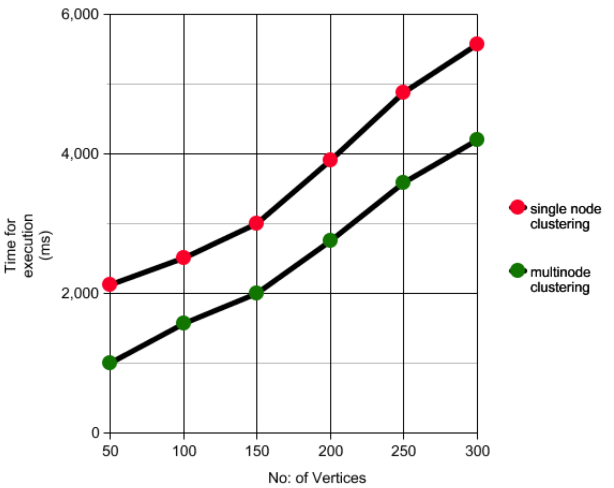

Figure 6: Comparison of results based on execution time when both concepts are implemented in single node and multi node hadoop cluster.

exchange between different people in a social network.Using Brandes algorithm betweenness centrality is computed and graph coloring is performed based on degree and betweenness centrality. Key nodes are those which have higher vertex values. Method is compared by implementing in 8GB and 16GB RAM single and multinode hadoop cluster. By including inbetweenness the solution becomes more optimal since the number of supersteps and number of colors are reduced in single node haddop cluster. Along with all constraint satisfaction in single node, execution time is also decreased in multi node hadoop cluster.

The algorithm proposed for graph coloring with inbetweenness is not flexible in nature therefore there are lots of ways in which the system can be improved which can be considered as future work. The proposed algorithm follow greedy approach, there is scope to improve the algorithm to a unique technique like parallel genetic,ant colony or watermarking algorithms.

References

[1] Nishant M Gandhi,Rajiv Misra, "Performance Comparison of Parallel Graph Coloring Algorithms on BSP Model using Hadoop", *International Conference on Computing, Networking and Communications, Cloud Computing and Big Data*, 2015.

[2] Andrew Lim and Fan Wang, "Meta-heuristics for Robust Graph Coloring Problem", *IEEE*, 2004.

[3] Tai-Wen Yue,Zou Zhong Lee, "A Q'tron Neural-Network Approach to Solve the Graph Coloring Problems", *19th IEEE International Conference on Tools with Artificial Intelligence*, 2007.

[4] Malika Bessedik, Rafik Laib, Aissa Boulmerka et Habiba Drias, "Ant Colony System for Graph Coloring Problem", *International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 2005.

[5] Sundar Vishwanathan, "Randomized Online Graph Coloring ", *IEEE*, 1990.

[6] Raja Marappan,Gopalakrishnan Sethumadhavan, 'Solution to Graph Coloring Problem using Divide and Conquer based Genetic Method", *International Conference On Information Communication And Embedded System*, 2016.

[7] Kamalpreet Singh, Ravinder Kaur, "Hadoop: Addressing Challenges of Big Data", *IEEE*, 2014.

[8] Hao Lu, Mahantesh Halappanavar, Daniel Chavarrıa-Miranda, Assefaw H. Gebremedhin, "Algorithms for Balanced Graph Colorings with Applications in Parallel Computing", *IEEE Transactions On Parallel And Distributed Systems*, 2017.

[9] TAKUYA AOKI, CLAUS ARANHA, HITOSHI KANOH, "PSO Algorithm with Transition Probability Based on Hamming Distance for Graph Coloring Problem", *IEEE International Conference on Systems, Man, and Cybernetics*, 2015.

[10] Isabel Mendez Diiaz, Paula Zabala, "A Generalization of the Graph Coloring Problem", *UBACYT Grant TW82*.

[11] "A Generalization of the Graph Coloring Problem - Supplementary notes", 2008.

[12] Fan Jiang,Carson K. Leung,Dacheng Liu,"Efficiency Improvements in Social Network Communication via MapReduce",*IEEE International Conference on Data Science and Data Intensive Systems*, 2015.

[13] https://scielo.conicyt.cl/scielo.php?script=sci$_{a}rttextpid = S0718 -$ 18762014000100006.

[14] https://en.wikipedia.org/wiki/Social$_{n}etwork_{a}nalysis/media/File$: $Graph_{b}etweenness.svg$.

[15] https://snap.stanford.edu/data/.