International Journal of Advance Research in Science and Engineering
Volume No 06, Special Issue No. (02), September 2017, ICITTESE-17
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Allotment of Cluster for different Nodes in Graph Computing

K. Ravikant[1], Gargi Shankar Verma[2]

[1, 2]*Department of CSE,Columbia Institute of Engineering & Technology, Raipurl*

ABSTRACT

In this paper, we will discuss the technique for finding the appropriate cluster group for the nodes when the number of cluster(k) is limited. This process is carried out by using k-means algorithm and BFS-Dijkstra, also named as k-dijkstra. Grouping of nodes in different clusters is easy when the value of k is given by the user. The system is very efficient when the clusters are less in number. So the problem arises when the number of nodes are high, shattered and the number of clusters are less. Here we will study the different techniques and issues faced with clustering.

Index Terms- K-Means, BFS, Dijkstra algorithm, K-Dijkstra.

I. INTRODUCTION

Clustering is the techniques used in data mining for extracting useful information from large amount of data stored in different databases and data warehouses. Clustering is used for grouping the set of data objects into groups or clusters so that objects within one cluster have high similarities between them but highly dissimilar from objects in other clusters. Let's take some examples of clustering for grouping the people of a city according to their age. grouping of students according to their branches of study. K-means algorithm in clustering is used to find the distance between the nodes or objects and the cluster centroids. The distance is calculated by either Euclidean distance, Manhattan distance or Minkowski distance. The grouping of objects is done on the basis of distance between the object and the cluster centroids.

In graph computing BFS(Breadth First Search) is used to traverse through different paths of the graph. BFS traverses the complete from level by level, so that the complete graph is checked and maintained. Dijkstra algorithm is used to find the distance of the nodes which are connected to the parent node. With the use if Dijkstra algorithm we can find the shortest path from source node and the destination node. Another technique was used to find the cluster groups known as k-dijkstra. It is used to find the cluster groups when the number of clusters(k) is not defined by the user or the system.

II. METHODS USED IN PRESENT SYSTEM

Let's take a group of objects in graph which needs to be clustered in different groups. Before analysis the graph,

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

2.1 BFS (Breadth First Search)

BFS is very much indispensable investigation strategies for node expansions. It traverses in a breadthward motion and uses the queue data structure to remember all the nodes of the graph. The rules of BFS are:

Rule 1: visit the adjacent unvisited node, mark it as visited and insert the node into the queue.

Rule 2: if no adjacent unvisited node found, remove the first node from the queue.

Rule 3: repeat rule 1 and rule 2 until queue is empty.

2.2 Dijkstra Algorithm

This algorithm is used to find the shortest path between a source and destination nodes. It is also known as single source shortest path because it has only one source node. It calculates the shortest path from source node to every un-visited node in the graph. Dijkstra algorithm can be used to find the shortest route between one city and all the other cities if the node represent the cities and the edges represents the driving distance.

2.3 K-Means Algorithm

K-means clustering intends to partition 'n' number of observations into 'k' number of clusters in which each object or node belongs to the cluster which is having the nearest mean known as centroid. In this method the inter-cluster similarity is very low and intra-cluster similarity is very high. The algorithm consist of two steps

Step 1: select 'k' randomly, where the value of k is fixed.

Step 2: each node or object in data set is related to the nearest centroid.

III. LITRATURE REVIEW

To lower the complexity of k-means clustering algorithm, Zhang Anna et al[1] proposed an algorithm which combines the features of BFS and Dijkstra algorithm. This algorithm was named as K-Dijkstra.

K-Dijkstra: Working methodology

To accelerate clustering, different techniques from k-means to collect vertices into different groups or clusters. The new concept is to divide the nodes into different zones according to the distance. Firstly the algorithm starts using dijkstra algorithm as shown in Fig.2 to find the distance between two nodes or every vertices. Now the dijkstra algorithm is implemented in a BFS structure and can easily be exploited using thread-level parallelism.

The algorithm of BFS-Dijkstra has the following steps:

Randomly pick one vertices as the root node and set it as a frontier vertex in CLASS[0]. Another array of CLASS[X] holds vertices distanced X to the current frontier node.

Check all the neighbours of the vertex in the current frontier array. Add each of the neighbours to the array CLASS[X], assuming the edge weight is always 1.

Repeat step2 until no new unvisted vertex left.

International Journal of Advance Research in Science and Engineering
Volume No 06, Special Issue No. (02), September 2017, ICITTESE-17
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

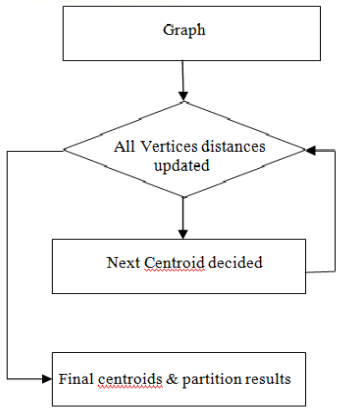

Fig 1: K-Dijkstra Flowchart

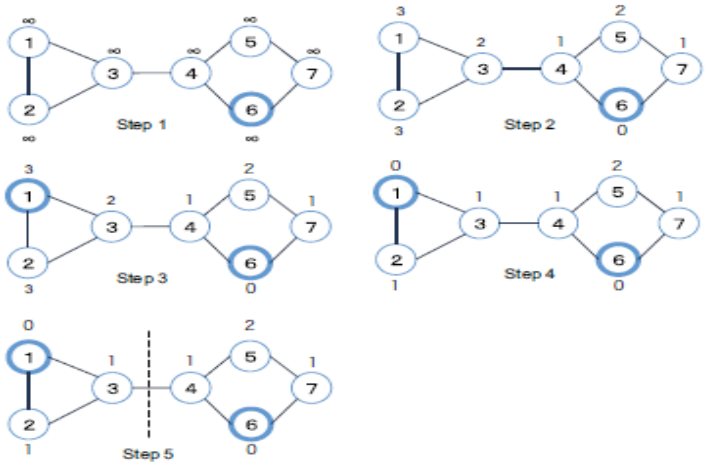

Fig 2: Example of K-Dijksta

3.1. *Deciding the centroids using k-dijkstra*

As the algorithm starts, it randomly selects one of the vertices as the root node or the frontier vertex as shown in Fig.3. The first centroid is selected which has the largest distance from the root node.

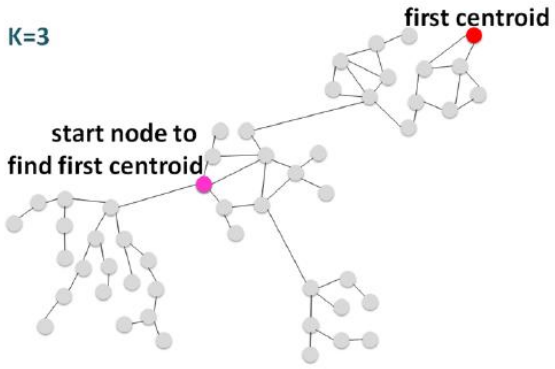

Fig 3: Find first centroid using k-dikstra

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Now the root node is changed to the first centroid which is selected by the algorithm in Fig.3. Again second centroid is decided using the algorithm in Fig.4.

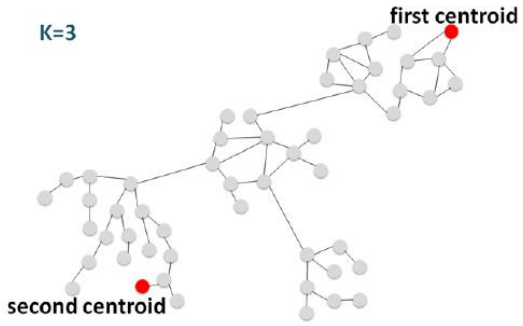

Fig 4: Find the next centroid

Again the root node is changed to the second centroid as in Fig.4. The value of K=3, therefore it requires another centroid for finishing the cluster.

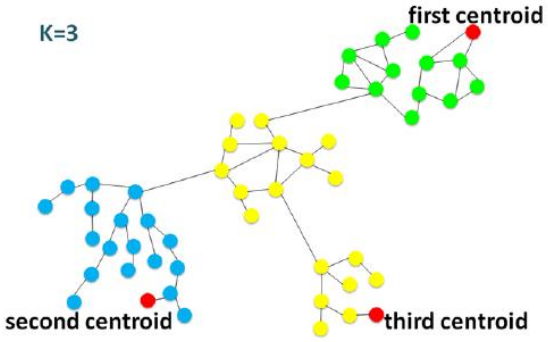

Fig 5: Last centroid decided and finishing the cluster.

3.2. *Failure of k-dijkstra*

In the above example graph having the number of clusters k=3, the algorithm finished with a good result. But if the cluster value is reduced to 2 than this algorithm fails. Fig.6 shows the failure if K=2. The nodes present in the middle of the graph are distributed into both the clusters. This result is true but not efficient.

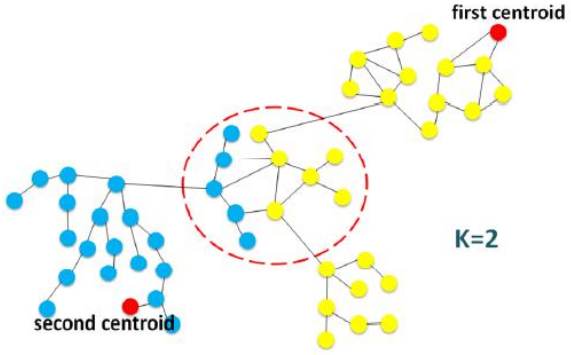

Fig 6: Cluster result in some bad situations.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

In Fig.7 another condition can be achieved with the above graph having k=2, which is better than Fig.6 but not realized yet.

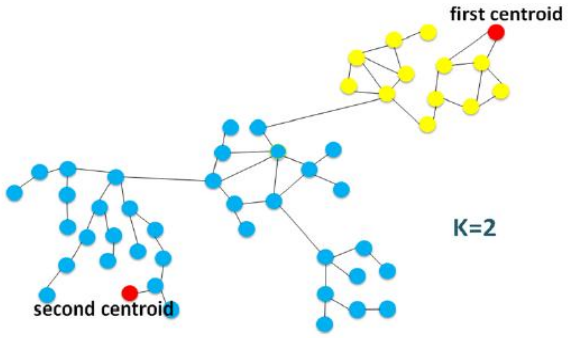

Fig 7: Better result not realized yet

IV. PROPOSED METHODOLOGY

If a condition occurs when the distance of the center nodes or vertices are same from first cluster and second cluster than in which cluster it will be grouped. The method shown in Fig.7 is permitted but not efficient.

The methodology which we are proposing for the problem identified in fig.7 is of two types:

1. Finding the distance between the connection nodes of respective clusters and the ungrouped cluster node. In Fig.8 the distance between AB and CD is calculated using Euclidean distance or Manhattan distance. As we can see the distance between AB is 3 and between CD is 5, so the nodes are grouped into second cluster.
2. With the same graph and distance calculation, if we found the distance between AB and CD are same than there is a problem that the center nodes will grouped into which cluster. Fig.9 shows this problem.

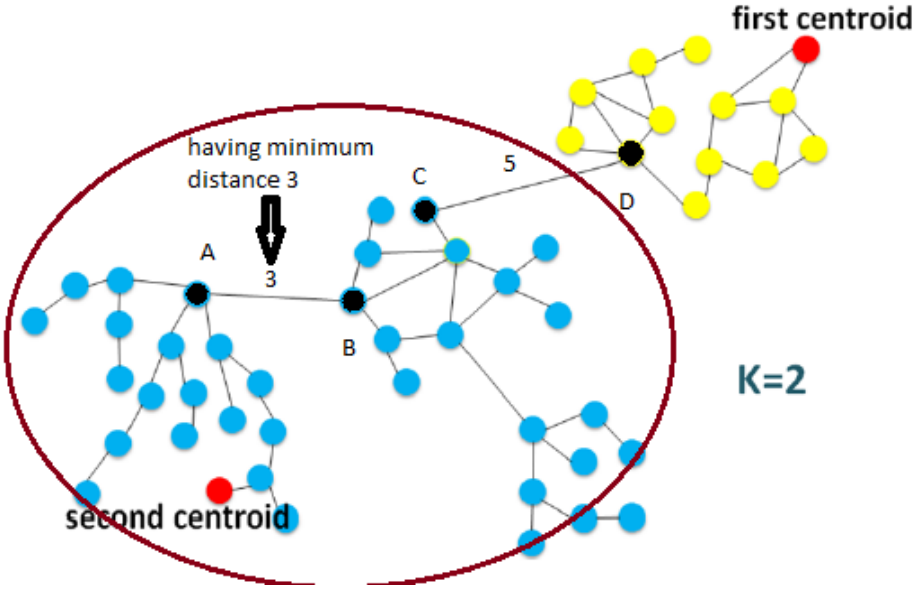

Fig 8: Connecting nodes with distances.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

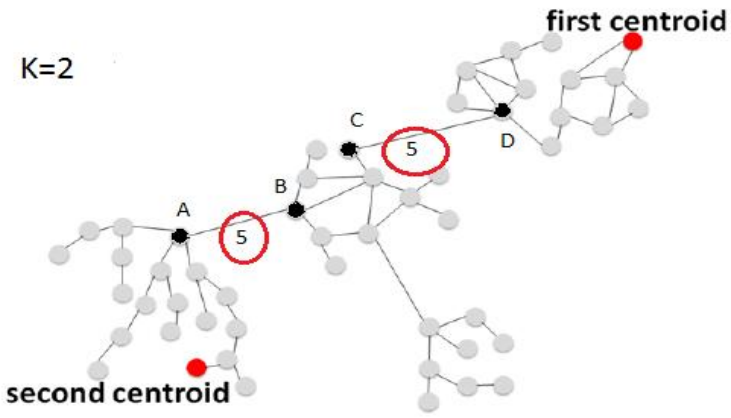

Fig 9: Connecting nodes with same distances.

Proposed methodology for the above problem is shown Fig.10, can be solved by dividing the entire graph into regions. The line of separation will overlap many edges from one node to another node of different regions.

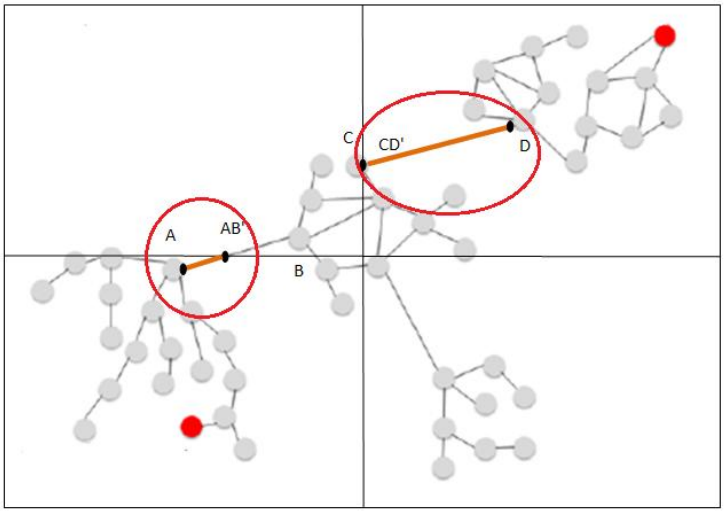

Fig 10: Graph divided into regions

Once the graph is divided we will find the intersecting points of the connected nodes, ie between AB and CD. The intersecting points are AB' and CD'. Now we will find the distance between A and AB', and similarly between D and CD'. The edge having minimum weight towards the cluster are grouped into that cluster. By the figure above we can analyze that the distance between A and AB' will be minimum, therefore the center nodes will be grouped into second cluster.

V. CONCLUSION

In this work we have studied about the k-dijkstra algorithm which divides the graph into regions according to the farthest distance of the vertices. After dividing the graph it is programmed into BFS style so that it can easily be traversed to find the right cluster group. It will help to find the right cluster groups when the number of clusters are minimum.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

REFERENCES

[1] Zhang Anna, Jun Yao, Yasuhiko Nakashima,"Lowering the complexity of k-means clustering by BFS-Dijkstra method for graph computing",IEEE Symposium on Low power and high speed chips 2015, April 2015.

[2] Aastha Jodhi, Rajneet kaur, "A Review: Comparative Study of Various Clustering Techniques in Data Mining" International Journal of Advanced Research in Computer Science and Software Engineering(IJARSSE), Volume 3, Issue 3, March 2013, pp. 55-57.

[3] Avinash Kaur, Purva Sharma, Apurva Verma,"A appraisal paper on Breadth-First Search, Depth-First Search and Red Black Tree", International Journal of Scientific and Research Publications,Volume 4, Issue 3, March 2014.

[4] Vaibhav Patel, Chitra Baggar,"A Survey paper on Bellman-Ford Algorithm and Dijkstra Algorithm for finding Shortest Path in GIS applications", International Journal of P2P Network Trends and Technology(IJPTT), Volume 5, February 2014.

[5] Jyoti Yadav, Monika Sharma,"A Review of K-Mean Algorithm", International Journal Of Engineering Trends and Technology(IJETT), Volume 4, Issue 7, July 2013,pp 2972-2976.

[6] Aniket Chandak, Rutika Bodhale, Raveena Burad," Optimal shortest path using HAS, A star and Dijkstra algorithm," imperial journal of interdisciplinary research (IJIR), Volume 2, Issue 4,2016.