THE UNIVERSITY OF SYDNEY

School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - INDIVIDUAL ASSESSMENT

Unit of Study: COMP5048 Visual Analytics

Assignment name: Construct a good visualisation of four graphs

Tutorial time: Thursday 8 p.m. to 9 p.m., weeks 1-13

Tutor name: Dr. Quan Nguyen

DECLARATION

I declare that I have read and understood the *University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy*, and except where specifically acknowledged, the work contained in this assignment/project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the the *Academic Dishonesty and Plagiarism in Coursework Policy*, can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

Student ID: 470374652

Student name: LINGZUO ZHAO

Signed Lingzuo zhao **Date** 2017.9.10

SIT Building, J12
The University of Sydney
NSW 2006 Australia

T +61 2 9351 3423
F +61 2 9351 3838
E sit.info@sydney.edu.au
sydney.edu.au/it

ABN 15 211 513 464
CRICOS 00026A

Picture of graph 1：Graph Drawing Citations

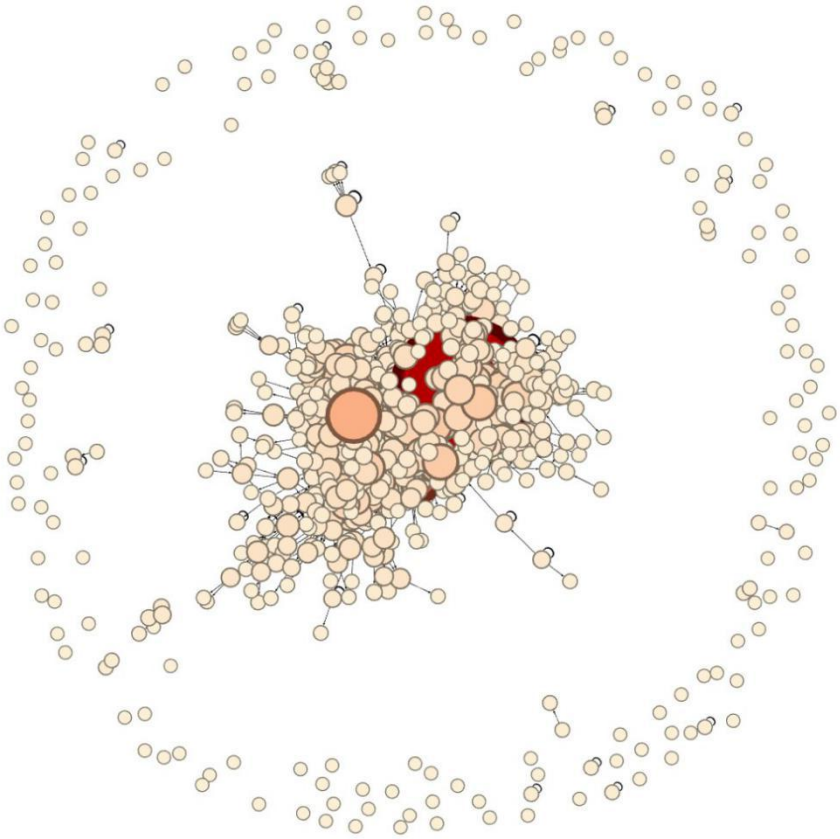

Figure1　overview of Graph Drawing Citations

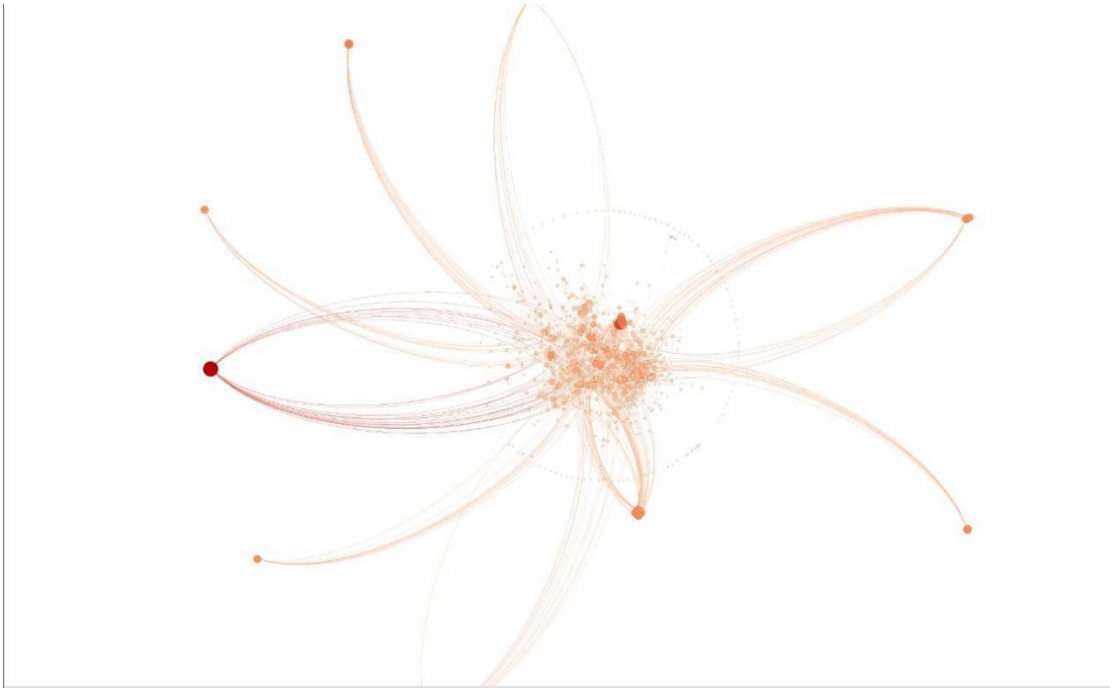

Figure2 pull some nodes out of the central area

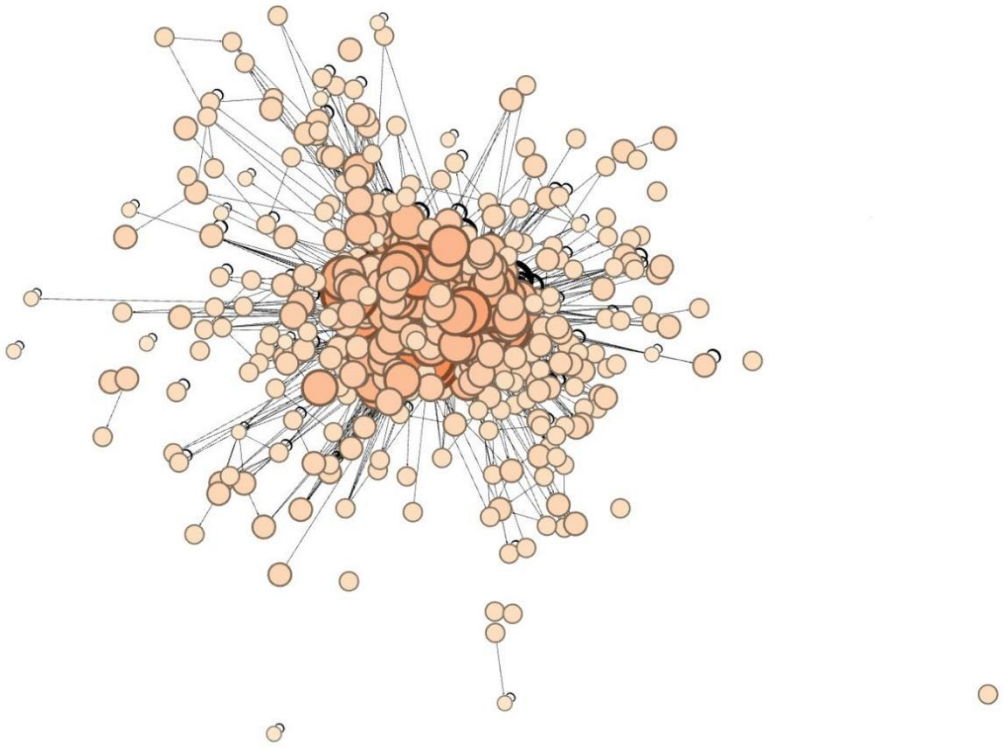

Figure3 Filtered with degree larger than 3

Description of graph 1

Graph creation: Graph Drawing Citations

Step1, processing the data

1. the data is downloaded through http://www.graphdrawing.de/contest2017/topics.html, it is a xml file.

2. transform the xml to json using online converter tool: http://www.utilities-online.info/xmltojson/#.WbM_E8gjHIU

3. using mongodb to process the data to get the edge.csv and node.csv

    1. create database,collection “a” in mongodb;

    2. import the json file to mongodb using mongoimport;

    3. to get the citedby relation, using $unwind to flatten the citedby array in each document, then $project the id and citedby fields to source and target fields respectively, $out to a new collection”b”. then export the source and target fields of collection “b” to a csv file using mongoexport tool. The same thing with cites relation. The two csv files are merged together in the microsoft excel, duplicate rows are deleted, saved as “edge.csv”.

    4. to get the degree of each node, operate with the collection “a”. using $lookup to join collections “a”, localField is cites, and the foreignField is id. Then using $size to calculate the number the paper

cites. For the number of the paper citedby, the process is the same. Then $add them together and $project to a new field “weight”, then export the “weight”, “id”, and” title” fields to a new csv file “node.csv” using mongoexport.

Step 2 draw the picture using Gephi

1. new a project in the Gephi
2. import the edge.csv and node.csv to the data laboratory
3. Then in the overview interface, select Yifan Hu algorithm, using default properties, run it.
4. Make the node’s color and size different with the weight’s difference, the nodes with more weight(that is the degree of the node) look bigger and darker.
5. Using degree filter to get the picture with nodes’ degree in a specified range.

Evaluation

There are some good points with this picture:

1. using good algorithm.

The selected Yifan Hu algorithm is one of the force-directed algorithms, it combines another two algorithms: the Force Atlas and Frunchterman Reingold algorithm. It is much more faster, only about 200 iterations to finish because of the way it optimizes the overall internode repulsions in the network. [1]

2. the information is more clearer

Because the more the paper citedby others and cites others, the more useful and more valuable the paper is. So with darker and bigger nodes representing the paper with more citedby and cites, the picture can convey clear information about which papers are more valuable.

3. with Gephi, the nodes of the picture can be easily analyzed, nodes with different degrees can be identified using filters.

Some limitations about this picture:

1. There are more than 1000 nodes, and more than 5000 edges, so the central part of the overview picture are very dense, and it is difficult to see the relationships between papers in the central part. Maybe using interactive tools can help users to explore the relationship.

Picture of graph 2：Composers Graph

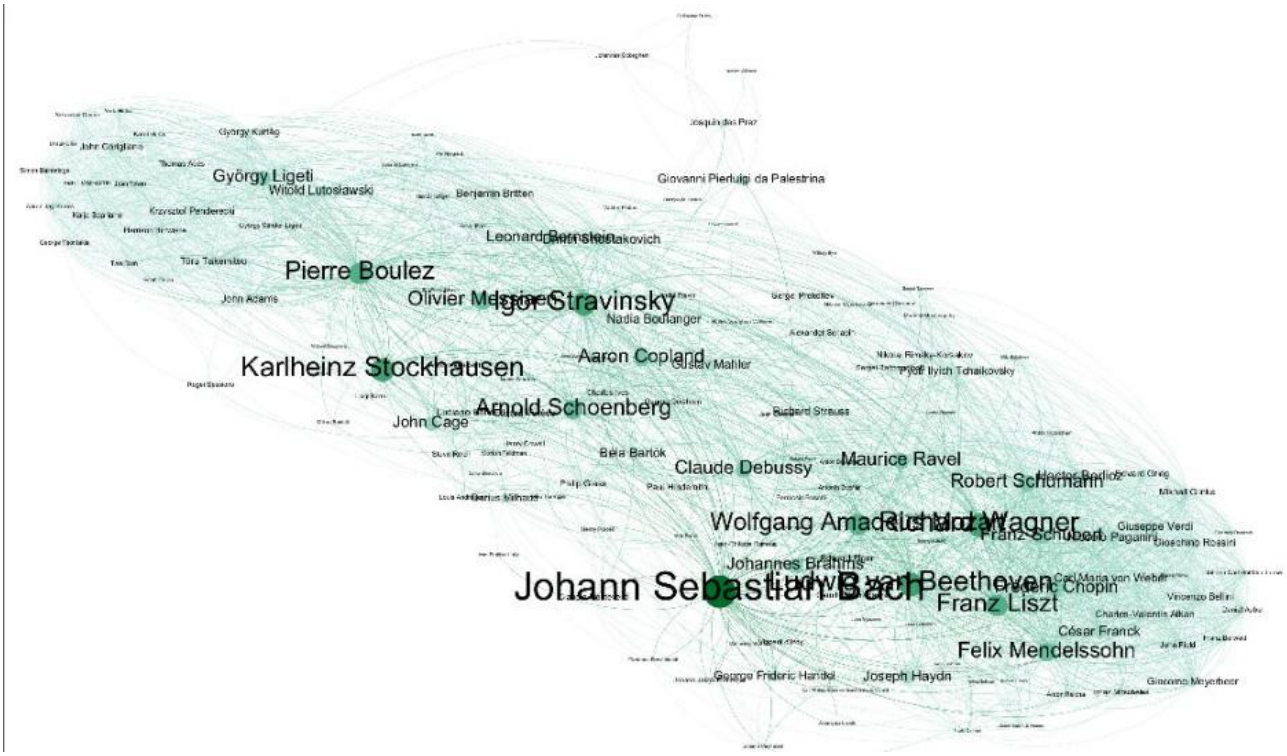

Figure4　overview of Composers Graph

Description of graph 2

Graph creation: Composers Graph

Step1, import the data to mongodb

1. the data is downloaded through http://www.graphdrawing.de/contest2014/topic2.html, it is a xml file.
2. transform the xml to json using online converter tool: http://www.utilities-online.info/xmltojson/#.WbM_E8gjHIU
3. separate the json file into two jsonArray(nodes and edges) in notepad++, then import them to .two collection in one database in mongodb using mongoimport

Step2, process the data in mongodb to get the edge.csv and node.csv of the most important 150 nodes

1. To get node.csv

For edge part's collection, $group according to source （which is a node's id） and target （which is also a node's id）field respectively, and $sum the total number of occurrences of each node as the source node and as the target node respectively, then $add them for each node, so each node's degree is calculated. Then using $sort (descending direction）and $limit, select top 150 nodes. The top 150 nodes is selected in the node

part's collection and $out to a new collection "nodes", $project the "id", "name","degree" fields to "id","label","weight", then export them to a new node.csv.

2. To get edge.csv

the edge collection join the top 150 nodes collection by $lookup, localField is the source(which is a node's id), foreignField is the id in the top 150 nodes collection, this operation will generate a new array field in the edge collection which contains source node information. So for each document in the edge collection, if the source is one of the top 150 nodes, then the new array field's size larger than zero. Based on this, those edges with source nodes which are not part of top 150 nodes are filtered. Then in the same way, those edges with target nodes which are not part of top 150 nodes are filtered. So the edge only between the top 150 nodes are selected, and export to the edge.csv using mongoexport tool.

The criteria to select the most important 150 nodes: for the total number of occurrences of the node as source and target, the higher ones are the more important ones.

Step 3 draw the picture using Gephi

1. new a project in the Gephi

2.import the edge.csv and node.csv to the data laboratory

3. Then in the overview interface, select ForceAtlas algorithm, adjust the parameters: Scaling:1000, and gravity:0.01,prevent overlap, then run it.

4. Make the node's color and size different with the weight's difference, the nodes with more weight(that is the degree of the node) look bigger and darker.

5. The label is the composer's name, show them on the node, the label's size is set according to the node's size which is set according to the degree of the node.

6. Rotate and zoom, and adjust the position of the picture, observe the result and adjust some nodes and edges.

Evaluation

Because there are only 150 nodes and the 2500 edges between them, so the ForceAtlas 2 is chosen. The picture clearly show the composer's name, more important composers' name looks more bigger, and the nodes representing them are more bigger and darker, it gives the user more clearer information about the more important composer and their relationships.

There are some limitations of this picture: the nodes and edges are not evenly positioned, and there are many bent edges. After selecting the algorithm and run it, how to use the software drag and pull some nodes to make the picture look good is hard to answer because there are so many edges here. When some nodes are moved, the nodes linked to them may have other edges to the rest rodes, so it is hard to see clearly how to move the nodes and the edges to make the visualization look better.

Picture of graph 3：Graph Classes

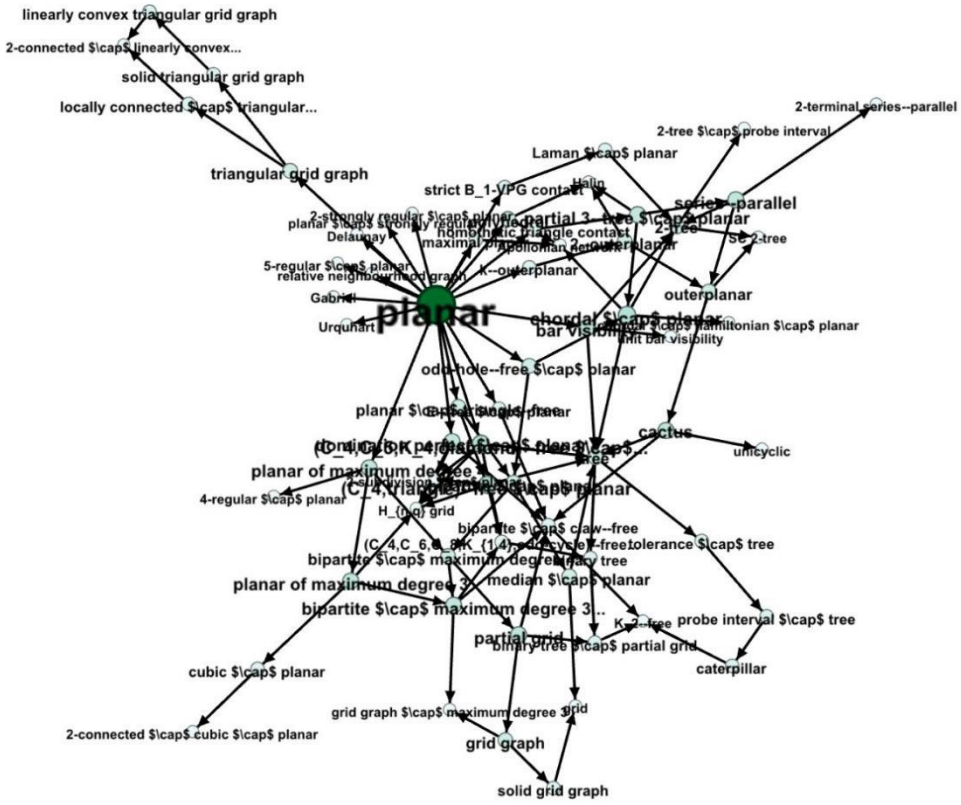

Figure5　Graph Classes with node label

Description of graph 3

Graph creation: Graph Classes

Step1, process the data

1. the data is downloaded through http://www.csun.edu/gd2015/planar.graphml, it is a xml file.

2. transform the xml to json using online converter tool: http://www.utilities-online.info/xmltojson/#.WbM_E8gjHIU.

3. separate the json file into two jsonArray(nodes and edges) in notepad++, then import them to .two collection in one database in mongodb using mongoimport.

4. modify nodes's fields as id, label using $project in the nodes collection, and modify edges's fields as source, target in the edges collection.

5.export id, label fields of nodes collection to nodes.csv ; export source,target fields of edges collection to edges.csv using mongoexport tool.

Step2: draw the picture in Gephi

1. new a project in the Gephi

2. import the edges.csv and nodes.csv to the data laboratory

3. Then in the overview interface, select Yifan Hu algorithm, using different properties to run it and observe the result.

4. Make the node’s color and size different with the out- degree, the nodes with more out- degree(which is more higher in the hierarchy) look bigger and darker. .

5. adjust the size of edges in the panel, make them more clearer

6. set the labels’ size according to the node’s size.

Evaluation

The graph contain the label of the vertices, and give a relatively good overview on the hierarchy of the graph classes. The size and color clearly show the category’s hierarchy, and there is a straight line from the top category to bottom category which make the user more clearly learn the route about the refinement of the category.

Some limitations:

The graph’s left part is slightly empty and middle looks crowded, it can be improved with more time and effort to drag and pull nodes and adjust edges’ length to make the visualization better.

Picture of graph4 ： Panama Papers

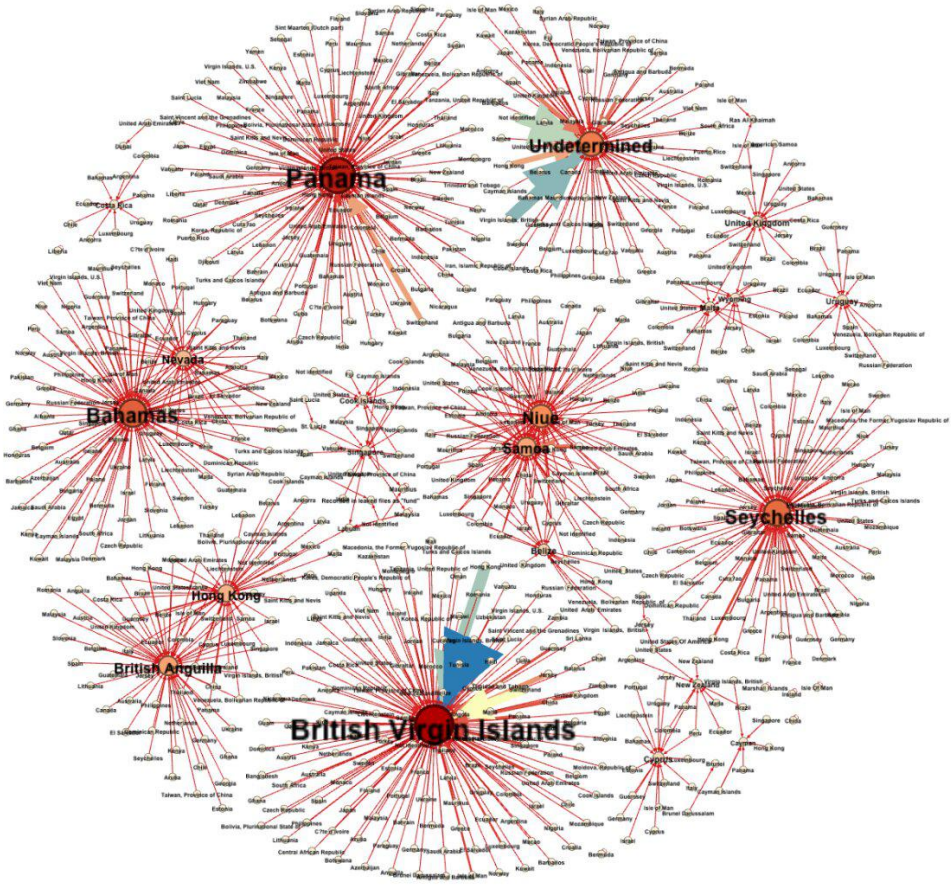

Figure6　Overview of Panama Papers

Small **big**

Legend: the edge's weight

Description of graph 4

Graph creation: Panama Papers

Step1, process the data

Download the data from

http://www.graphdrawing.de/contest2016/panama.txt.

Process the data using notepad++ and microsoft excel, form a edge.csv which has source, target, weight.

Step2, draw picture by Gephi

1. import the edge.csv into data laboratory.
2. The nodes are automated extracted from the edges. Copy the id column to label column in the nodes table.
3. select the Fruchterman Reingold layout, and run it.
4. Make the node's color and size different with the in-degree, the nodes with more in- degree look bigger and darker.
5. set the labels' size according to the node's size
6. Set the edges' color according to the edges' weight.

Evaluation

The graph contains all names of the countries and gives a relatively good overview on their correlation.And all the target countries' names and the nodes representing them look bigger and darker, it is easy for users to identify the target countries first and find its source countries.Further more the edges' color reflects the edge's weight, it is clear for users to notice which source countries link more entities in the target countries.

Some limitations: the source countries that link to many target countries are not clearly shown through this picture.

References

[1] Using the Yifan Hu layout algorithm.[ONLINE] Available at: https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781783987405/3/ch03lvl1sec41/using-the-yifan-hu-layout-algorithm. [Accessed 10 September 2017]