An Algorithmic Solution for the "Hair Ball" Problem in Data Visualization

Khalid H. Alnafisah
Department of Computer Science
Louisiana State University
3325E Patrick F Taylor Building, Baton Rouge. LA 70803
alnafisah.k.h@ieee.org

1. Abstract

The investigation and analysis of large and complex graphs is an important aspect of data visualization research, yet there is a need for entirely new, scalable approaches and methodologies for graph visualization. This can ultimately provide more insight into the structure and function of this complex graph (Hair Ball). To explain more, we need to find a methodology to develop a solution to present a "Tidy" graph with the minimal crossover between edges in the "Hair Balls." In spite of the expanding significance of investigating and extensively analyzing and understanding very large graphs of data, the traditional way of visualizing graphs has difficulties scaling up, and typically ends up depicting these large graphs as "Hair Balls". This traditional approach does indeed have a deeply intuitive foundation: nodes are depicted with a shape such as a circle, triangle or square, which are then connected by lines or curves that represent the edges. In any case, although there are many different ways to apply this basic underlying idea, it needs to be revisited in light of current and emerging needs for understanding increasingly complex crossover between edges in the graphs. The complex "Hair Ball," which appears as an indecipherable graph, came from the crossover between edges. From our preliminary research, we found the major disadvantage in the Hair Balls graph was that it confused observers. Users may think there are some extra nodes; but in reality, there are not. Because there are many crossovers between edges in the Hair Balls, the impression also may affect observers' understanding of the whole structure of the graph. Major problem-no effective reception of information from a "Hair Balls" graph-meaningless to observers.

2. Keywords

Bipartite $K_{2,3}$, Crossover, Algorithmic, Complex Graph.

مفاتيح الكلمات

ذو قسمين ك ٢٣, تقاطع, لوغاريتم, رسم بياني معقد

حل لوغاريتممي لمشكلة "كرة الشعر" في تصوير البيانات

الملخص

يعتبر تحليل الرسوم البيانية الضخمة والمعقدة وتحليلها جانباً مهماً من جوانب بحوث التصور ، ومع ذلك هناك حاجة إلى مناهج ومنهجيات جديدة قابلة للتطوير كلياً للتصور البياني. يمكن لهذا أن يوفر في نهاية المطاف المزيد من التبصّر في بنية هذا الرسم البياني المعقد ووظيفته. وللتوضيح أكثر ، نحتاج إلى إيجاد منهجية لتطوير حل لتقديم رسم بياني "أنيق" مع الحد الأدنى للتقاطع بين الحواف في "كرات الشعر." على الرغم من الأهمية المتزايدة للتحليل على نطاق واسع وفهم الرسوم البيانية الكبيرة جدًا من البيانات، فإن الطريقة التقليدية لتصور الرسوم البيانية لها صعوبات في التوسع ، وعادة ما ينتهي الأمر بتصوير هذه الرسوم البيانية الكبيرة باسم "كرات الشعر." هذا النهج التقليدي له أساسًا بديهيًا للغاية: يتم تصوير العقد بشكل مثل دائرة أو مثلث أو مربع ، ثم يتم توصيلها بواسطة خطوط أو منحنيات تمثل الحواف. على أي حال ، على الرغم من وجود العديد من الطرق المختلفة لتطبيق هذه الفكرة الأساسية، إلا أنه يجب إعادة النظر فيها في ضوء الاحتياجات الحالية والناشئة لفهم التبادل المعقد بشكل متزايد بين الحواف في الرسوم البيانية. جاءت "كرة الشعر" المعقدة التي تظهر على شكل رسم بياني غير مرتب بين الحواف. من بحثنا الأولي ، وجدنا أن العيب الرئيسي في الرسم البياني لكرات الشعر هو أنه يعمل ارتبك للمراقبين. قد يعتقد المراقبين أن هناك بعض العقد الإضافية بين التقاطعات في كرت الشعر. لكن في الواقع ، لا يوجد. نظرًا لوجود العديد من عمليات الانتقال بين الحواف في كرات الشعر، قد يؤثر الانطباع أيضًا في فهم المراقبين للهيكل الكلي للرسم البياني والذي يعد مشكلة كبيرة مما يجعل الرسم البياني "كرات الشعر" بلا معنى للمراقبين.

3. INTRODUCTION

3.1. Overview

In this research, we studied the questions, “What is the Hair Ball Problem?” and “What is the reason for the Hair Ball Problem?” Discovering answers to these questions will lead observers to understand the “Hair ball “problem in data visualization. In large graphs, some crossovers between edges are always unavoidable, for example, Fig.1. We found that less crossovers in large graphs allows us to develop a solution (Complete Bipartite Graph $K_{2,3}$) to present a “tidy” graph (semi-tree structure). This problem is inspired by those recent “big data” research. The nature of this re- search will be about developing a new algorithm to provide a better layout for big graphs [Ortmann & Brandes, 2015] [Richard & David, 2013].

Fig. 1: Crossovers between edges

3.2 Identify the Problem Area

In large graphs, too many crossovers are unavoidable (Two edges cross each other on the graph presentation which may cause confusion for extra node) [Carter, 2008] [Ranjan, Banerjee, Dey, Ganguli & Sarkar, 2014]. In more explain, one cannot avoid multiple crossovers in large graphic data. As shown in figure 2, crossovers are classic data visualization problem. The disadvantage is the original graph may con- fuse observers [Tardos & Kleinberg, 2006]. It is difficult to observe the structure of the graph. In wider explain, disadvantage include graph confusion due to an overwhelming numbers of crossovers of crossovers, and an inability to decipher information [Hartung, 2006].

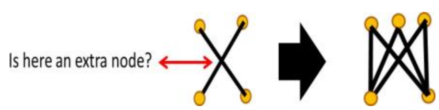

Fig. 2: Crossover in $K_{2,}$

As you see in figure 2, observers may not be able to see node behind this crossover.

4. APPROACH

4.1 Hypothesis

- The “hair ball” problem in data visualization is entirely dependent on $K_{2,3}$.
- $K_{2,3}$, a discrete solution can be applied to a “Big Graph”.

In this research, the algorithmic solutions will steps as follow [Gori, Maggini & Sarti, 2005]:

- **Step 1**: we find and locate all the $K_{2,3}$ graph within a complex graph.
- **Step 2**: by compressing the $K_{2,3}$, we can create a new graph.
- **Step 3**: we prune the new graph separate the $K_{2,3}$ sections (expand the $K_{2,3}$ nodes to discern data), and so we can see if the graph is tidy or has a complex crossover between the edges.

- **Step 4**: represent these $K_{2,3}$.

Note, the description above is only a rough idea. More complicated structure, such as loop and embedded func- tions may be used in the real solutions.

5. TECHNICAL APPROACH

5.1 R Script [Csardi, 2015] [Maris, 2008] or Gephi Visualization Tool [Heymann, Grand & Bndicte, 2013] [Egawa & Furuya, 2014]

In this part, I used R program and Excel to plot the graph. I used R programing because R is an integrated suite of software facilities for data manipulation, calculation, and graphical display.

6. IMPLEMENTATION DESIGN (ALGORITHM)

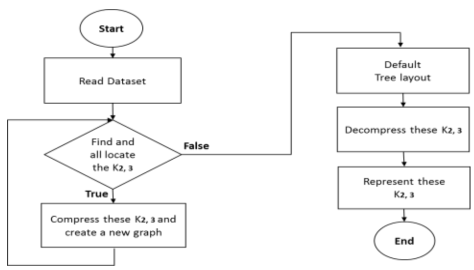

Fig. 3: Implementation design (Algorithm).

7. METHODOLOGY

7.1. Objectives

- Study the “Hair Ball” problem in data visualization.
- Develop a solution to present a tidy graph with minimal crossover between edges in the Hair Balls.
- Test, evaluate and verify my solution on multiple (at least three) big and complex graph datasets.

7.2. Experimental Design

7.2.1 Dataset

7.2.1.1 Small Datasets – There are four small dataset that we made-up it. These four small dataset with a $K_{2,3}$ or without any $K_{2,3}$.

7.2.1.2 Big Dataset – There are three big dataset we will use it in this research. These three big dataset from different resources / categories. At least 100 nodes for each graph.

7.2.1.3 Big Dataset -Media Organizations Dataset – It is a network of links between media venues and consumers. We will rarely use certain visual properties such as the shape of the node symbols and $K_{2,3}$ Bipartite: those are impossible to distinguish in larger graph maps. In fact, when drawing very big networks we may even want to hide the network edges, and focus on $K_{2,3}$ bipartite of nodes. At this point, the size of the networks you can visualize in R is limited mainly by the RAM of your machine.

7.2.2 Workflow of Small Datasets (One $K_{2,3}$ Bipartite)

An Excel file is used to create the dataset which have one $K_{2,3}$. Two columns are evident with seventeen rows in the Excel sheet. The name of the first column is "x" and the second column is called "y." I give the first three rows in the columns of "x" same number which is "1." Also, I give the second three rows in the columns of x same numbers but different than number "1" which is "2."

In the opposite direction of the column "y," I give the first three rows in the columns three different numbers which is "3, 4 and 5." Also, I will give the second three rows in the column of "y" the same group in the first three rows in column "y" which is "3, 4 and 5." The rest of the rows in both columns we will give those randomly numbers but completely different than which we gave to the first six rows in both column as showing in figure 4.

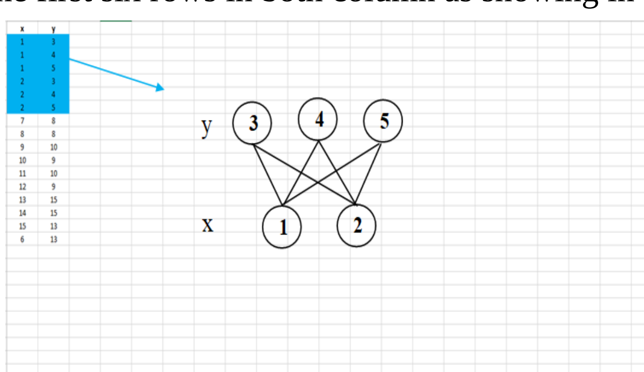

Fig. 4: One $K_{2,3}$ Bipartite.

The small dataset, there is one $K_{2,3}$ bipartite. By using the linkcomm package (with the library in R programing) you can check if the dataset has any $K_{2,3}$ Bipartite or not. But, before I use this package or any package I have to use the “igraph package with it library,” for a Network Analysis and Visualization.

The linkcomm package, is a “provide tool” for the gen- eration, visualization, and analysis of link communities in networks of arbitrary size and type [Kalinka, 2014]. As shown in figure 5, there is one $K_{2,3}$ in the dataset, because when we use the linkcomm command we will have 5 nodes in the cluster which is these 5 nodes in cluster are one $K_{2,3}$ bipartite.

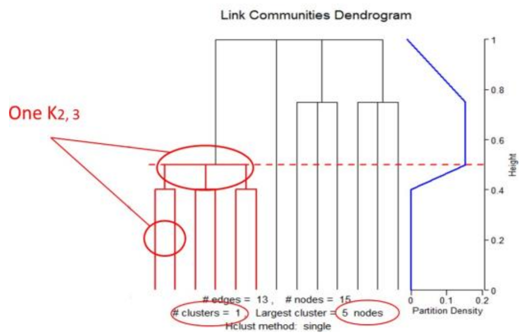

Fig. 5: One $K_{2,3}$ Bipartite by Linkcomm Package before Compress.

If the Linkcomm Package indicates a $K_{2,3}$ cluster, the next step would be to use the Bipartite Layout to calculate a variety of indices and values for a bipartite network [Katherine, 2015]. As shown in figure 5, there is one $K_{2,3}$ because we have one cluster with 5 nodes which confirms this cluster is One $K_{2,3}$ Bipartite [Etemad, Carpendaleya, & Samavatiz, 2014].

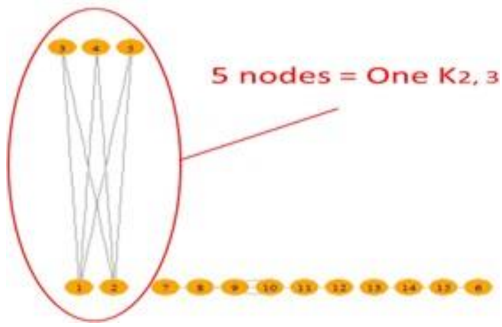

Fig. 6: One $K_{2,3}$ Bipartite by Bipartite Package before Com- press.

After confirming the K2,3 bipartite is ineligible for this dataset, we will use a special layout [Dunne & Ben, 2013] to make this graph a "tidy" graph. The first step is to see how this dataset looks (visualization) by using a random tree layout as shown in figure 6 [Katherine, 2015]

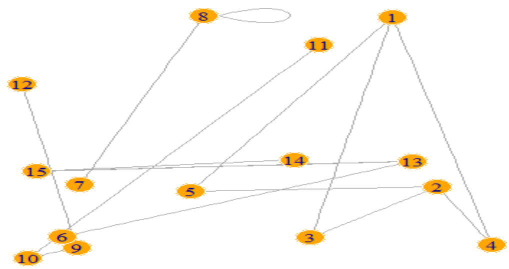

Fig. 7: Random Tree Layout.

Figure 7 is not a tidy graph, because it has one $K_{2,3}$. After we find and locate this $K_{2,3}$, compressing (cluster) the next step is to will makes this graph a tidy graph by compressing the $K_{2,3}$ and creating a new graph. The following procedure is to assign all the nodes in the $K_{2,3}$ to one letter or number to be come all these nodes (5 node in one cluster) in the same number or letter. The next step will assign all the nodes in the $K_{2,3}$ to B Followed by a number and different color around the node than others node, for example "B1" with yellow color around the node.

After that, we will plot this graph to find this $K_{2,3}$ become one node not 5 node as we saw in the Tree layout [O'HAIR, 2009] graph before we use this scenario, see figure 8.

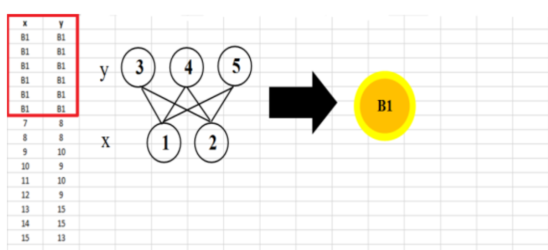

Fig. 8: Compress One $K_{2,3}$ Bipartite

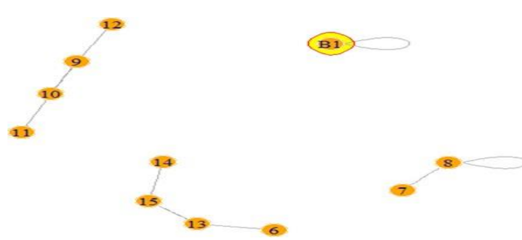

Fig. 9: Tree Layout after the Compress the $K_{2,3}$.

After compressing the $K_{2,3}$ and creating the new graph, the next step is to check if there ianother $K_{2,3}$ in the graph by using the Linkcomm Package feature as shown in figure 9 [Kalinka, 2014].

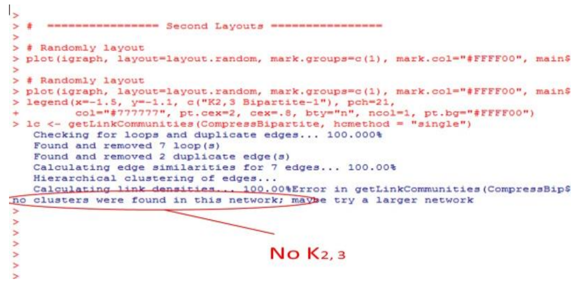

Fig. 10: No K by Linkcomm Package

As evident in figure 10, there are no more $K_{2,3}$ in the dataset, because all of them have been compressed. The figure indicates, “no clusters were found in this network”, which means there are no more $K_{2,3}$ in the dataset.

Verify the absence of $K_{2,3}$ bipartite, the next step is to use the bipartite package which calculates a variety of indices and values for a bipartite network. As shown in figure 11, there are no more $K_{2,3}$ because the nodes at the bottom cannot connect to nodes on the top of the figure.

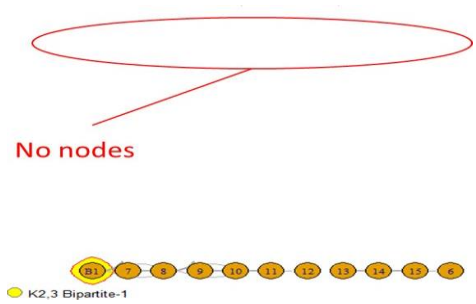

Fig 11: Bipartite Package after the Compress the $K_{2,3}$.

In this dataset there is one $K_{2,3}$ in the graph. Last step in the algorithm steps of the hypothesis is to use “Represent $K_{2,3}$” through R or Gephi. To represent this $K_{2,3}$ by R, just selected and customized one of the available layout algorithms in it as shown in figure 12 [Katherine, 2015].

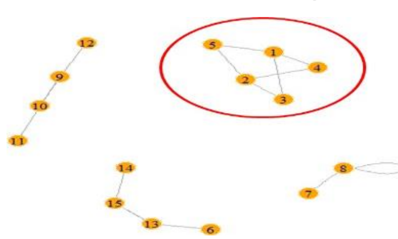

Fig. 12: Represent Graph by Tree Layout.

All the node in the interface of Gephi in the same color except one node. This one node (orange color) is one $K_{2,3}$. We compressed five nodes in one node. When we back to represent all the nodes, the only one it will be change is the node that has the $K_{2,3}$ (orange node) as shown in figure **9** and **10**.

8. VALIDATION

8.1. Small Datasets

8.1.1 Small Dataset (Five $K_{2,3}$ Bipartite)

An Excel file is used to create the dataset which have five $K_{2,3}$ bipartite. There are two columns with 37 rows in the Excel sheet. The name of the first column is “x” and the second column is called “y.” The first three rows in the columns of “x” same number which is “1.” Also, the second three rows in the columns of “x” same numbers are labeled “2.” In the column “y,” the first three rows in the columns three are labeled as “3, 4 and 5.” Also, the second three rows in the column of “y” the same group in the first three rows in column “y” which is “3, 4 and 5.”In the second $K_{2,3}$ in this dataset labeled give the third three rows in the columns of “x” same number which is “6.” Also, give the fourth three rows in the columns of x same numbers but different than number “6” which is “7.” In the opposite direction which is column “y,” We will give the third three rows in the columns three different numbers which is “8, 9 and 10.” Also, we will give the fourth three rows in the column of “y” the same group in the third three rows in column “y” which is “8, 9 and 10.” In the third $K_{2,3}$ in this dataset we will give the five three rows in the columns of “x” same number which is “11.” Also, give the sixth three rows in the columns of x same numbers but different than number “11” which is “12.” In the second column which is column “y,” we will give the five three rows in the columns three different numbers which is “13, 14 and 15.” Also, give the sixth three rows in the column of “y” the same group in the five three rows in column “y” which is “13, 14 and 15.”

In the fourth $K_{2,3}$ in this dataset we will give the seventh three rows in the columns of “x” same number which is “16.” Also, will give the eighth three rows in the columns of x same numbers but different than number “16” which is “17.” In the opposite direction which is column “y,” we will give the seventh three rows in the columns three different numbers which is “18, 19 and 20.” Also, we will give the eighth three rows in the column of “y” the same group in the seventh three rows in column “y” which is “18, 19 and 20.”

In the fourth $K_{2,3}$ in this dataset we will give the ninth three rows in the columns of “x” same number which is “21.” Also, we will give the tenth three rows in the columns of x same numbers but different than number “21” which is “22.” In the second column which is column “y,” we will give the ninth three rows in the columns three different numbers which is “23, 24 and 25.” Also, we will give the tenth three rows in the column of “y” the same group in the ninth three rows in column “y” which is “23, 24 and 25.”

Between these $K_{2,3}$ there is $K_{2,3}$. We chose one node from every $K_{2,3}$ that we explained in the previous to make a $K_{2,3}$ between these five $K_{2,3}$. We will give the eleventh three rows in the columns of “x” same number which is “17.” Also, we will give the twelfth three rows in the columns of x same numbers, but different than number “17” which is “22.” In the second column which is column “y,” we will give the eleventh three rows in the columns three different numbers which is “5, 10 and 15.” Also, we will give the tenth three rows in the column of “y” the same group in the ninth three rows in column “y” which is “5, 10 and 15.” This is the Scenario of the algorithmic on this dataset as shown in the figure 13.

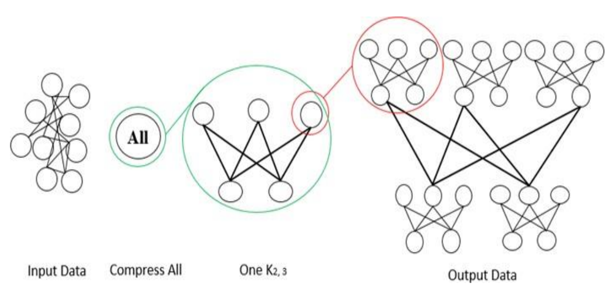

Fig. 13: Algorithm scenario on the dataset.

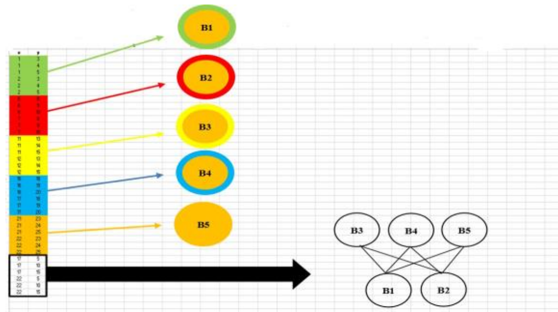

Fig.14: Five Bipartite $K_{2,3}$ in One Bipartite $K_{2,3}$.

In the small dataset, there are six $K_{2,3}$ bipartite. By using the linkcomm package (with the library in R programing) the prevalence of $K_{2,3}$ bipartite can be verified. Before this package, or any package, are used the "igraph package with it library," Network Analysis and Visualization. The linkcomm package is a "provide tool" for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. As shown in figure 14, there are six $K_{2,3}$ in the dataset (5 $K_{2,3}$ and one $K_{2,3}$ between those $K_{2,3}$), because when we use the linkcomm command, we will have six clusters each containing nodes. These 5 nodes in one cluster are one $K_{2,3}$ bipartite.

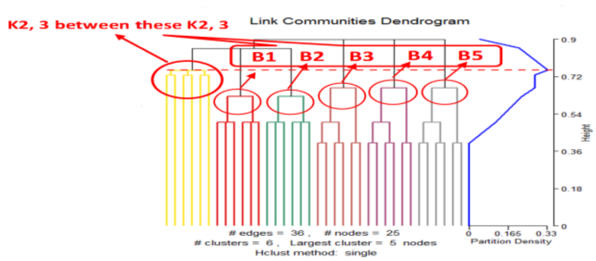

Fig. 15: Six $K_{2,3}$ Bipartite by Linkcomm Package before Compress.

If the Linkcomm Package indicates a $K_{2,3}$ cluster, the next step is to see how this dataset looks (visualization) by using a Random Tree Layout as shown in figure 16.

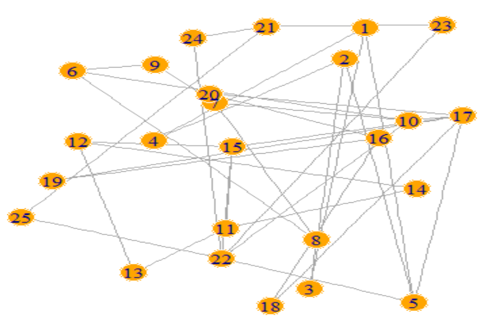

Fig. 16: Random Tree Layout.

Figure 16 is not a tidy graph, because it has six $K_{2,3}$. After we find and locate these $K_{2,3}$, compress (cluster) the next step is to makes this graph a tidy graph by compressing these $K_{2,3}$ and creating a new graph. The following proce- dure is to assign all the nodes in the $K_{2,3}$ to one letter or number to be come all these nodes (5 node in one cluster) in the same number or letter. The next step is to assign all the compressing nodes in the first $K_{2,3}$ to B followed by a number and different color around the compressing node than others node, for example “B1” with green color around the compressing node for the first $K_{2,3}$. The red is assigned to the compressing node for the second $K_{2,3}$ with the “B2.”

Yellow color is assigned to the compressing node for the third $K_{2,3}$ with the “B3.” Blue color is assigned around the compressing node for the fourth $K_{2,3}$ with the “B4.” Orange color is assigned to the compressing node for the $K_{2,3}$ with the “B5” as shown in figure 17 [Carter, 2008].

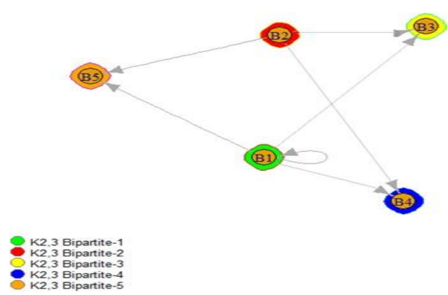

Fig. 17: Tree Layout-First Compress the $K_{2,3}$.

After compressing the $K_{2,3}$ and creating the new graph, the next step is to check if there is another $K_{2,3}$ in the graph by using the Linkcomm Package feature as shown in figure 18.

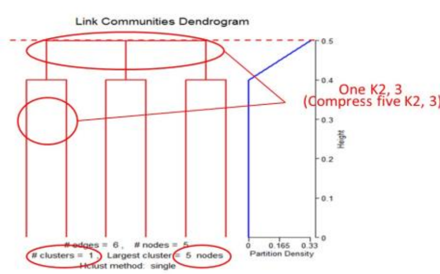

Fig. 18: Linkcomm Package-First Compress the $K_{2,3}$.

If the Linkcomm Package indicates a $K_{2,3}$ cluster, the next step would be to use the Bipartite Package to calculate a variety of indices and values for a bipartite network. As shown in figure 19, there is one $K_{2,3}$ because we have one clusters with 5 nodes for each which confirms these clusters are "One $K_{2,3}$ Bipartite."

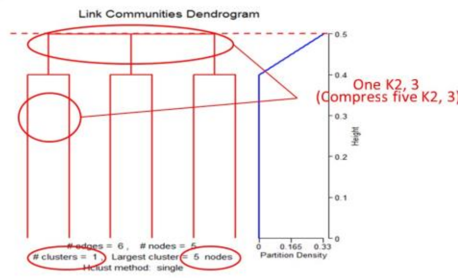

Fig. 19: Linkcomm Package-First Compress the $K_{2,3}$.

If the Linkcomm Package indicates a $K_{2,3}$ cluster, the next step would be to use the Bipartite Package to calculate a variety of indices and values for a bipartite network. As shown in figure 20, there is one $K_{2,3}$ because we have one clusters with 5 nodes for each which confirms these clusters are "One $K_{2,3}$ Bipartite."

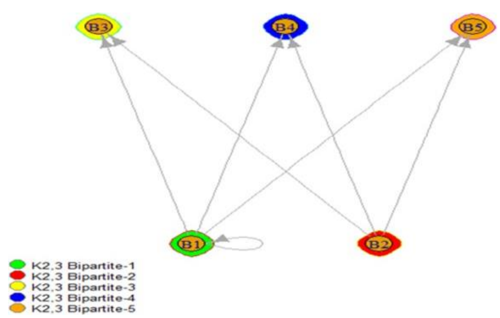

Fig. 20: Bipartite Package-First Compress the $K_{2,3}$. Figure 16 is not a tidy graph, because it has one $K_{2,3}$.

After we find and locate this $K_{2,3}$, compress (cluster) the next step is to makes this graph a tidy graph by compressing this $K_{2,3}$ and creating a new graph. The following procedure is to assign all the nodes in the $K_{2,3}$ to one letter, number or word to be come all these nodes (5 node in one cluster) in the same number, letter or word. The next step is to assign all the compressing nodes in the first $K_{2,3}$ to "All" with a color around the compressing node, for example, "All" with yellow color around the compressing node which meant this node call "All" is mean: B1, B2, B3, B4 and B5 as shown in figure 17.

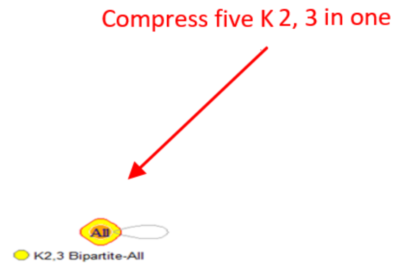

Fig. 21: Tree Layout-Second Compress the $K_{2,3}$.

After compressing the $K_{2,3}$ and creating the new graph, the next step is to check if there is another $K_{2,3}$ in the graph by using the Linkcomm Package feature as shown in figure 21. Before the last step (Represent the $K_{2,3}$ by Gephi), we will change the position of the node as shown in figure 22.

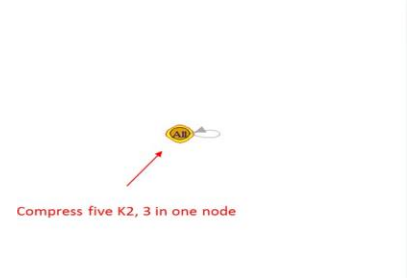

Fig. 22: Change the Position of the Node.

In this dataset there are five $K_{2,3}$ in one $K_{2,3}$ in the graph. The last part of the algorithm solutions of the hypothesis "Represent $K_{2,3}$" is performed through Gephi. To represent these $K_{2,3}$ by R, we selected and customized one of the available layout which is Tree Layout as shown in figure 23, 24 and 25.

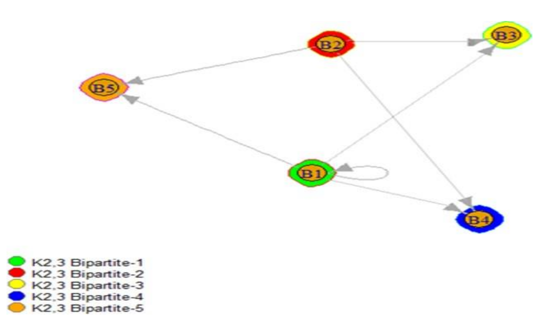

Fig. 23: First Represent Graph by Tree Layout.

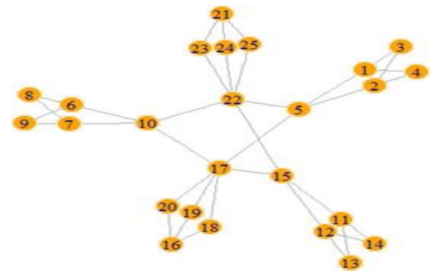

Fig. 24: Second Represent Graph by Tree Layout.

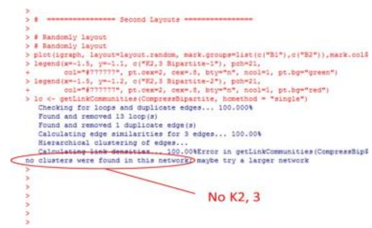

Fig. 25: No K by Linkcomm Package-Second Compress by Linkcomm Package-Second Compress

8.2 Big Datasets

8.2.1 Big Dataset -Media Organizations (14 $K_{2,3}$ Bipartite)

The Big datasets of Media Organizations in excel contains 14 $K_{2,3}$ Bipartite. Some of the initial $K_{2,3}$ are compressed into the $K_{2,3}$. a secondary K2,3. By using the linkcomm package with the library in R programing, the dataset can through checking. After confirming the K2,3 bipartite is ineligible for this dataset, we will use a special layout to make sure if this graph a “tidy” graph or not by using a Random Tree Layout as shown in figure 26.

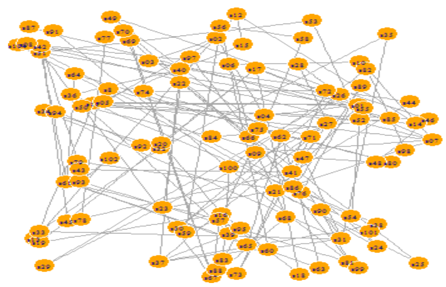

Fig. 26: Random Tree Layout.

Figure 24 is not a tidy graph, because it has 14 $K_{2,3}$. After we find and locate these $K_{2,3}$, compress (cluster) the next step is to makes this graph a tidy graph by compressing these $K_{2,3}$ and creating a new graph. The following proce- dure is to assign all the nodes in the $K_{2,3}$ to one letter or number to be come all these nodes (5 node in one cluster) in the same number or letter. The next step is to assign all the compressing nodes in the first $K_{2,3}$ to the letter B followed by a number and different color around the compressing node than others node, for example “B1” with a green color around the compressing node for the first $K_{2,3}$. Red color is assigned around the compressing node for the second $K_{2,3}$ with the “B2.” Yellow color is assigned around the compressing node for the third $K_{2,3}$ with the “B3.” Blue color is assigned around the compressing node for the fourth $K_{2,3}$ with the “B4.” Orange color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B5.” Cyan color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B6.” Firebrick color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B7.” Black color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B8.” Brown color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B9.” Magenta color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B10.” Tomato color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B11.” Pink color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B12.” Salmon color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B13.” Purple color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B14” as shown in figure 27.

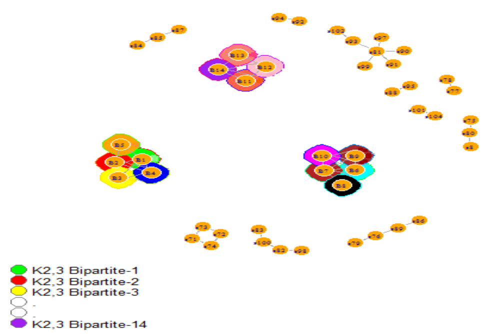

Fig. 27: Tree Layout of 14 $K_{2,3}$ (First Compress).

After compressing the $K_{2,3}$ and creating the new graph, the next step is to check if there is another $K_{2,3}$ in the graph by using the Linkcomm Package feature (loop) as shown in figure 25.

Figure 25 is not a tidy graph, because it has two $K_{2,3}$. After the $K_{2,3}$ is located, it needs to be compress to makes this graph a tidy graph by compressing this $K_{2,3}$ and cre- ating a new graph. The following procedure is to assign all the nodes in the $K_{2,3}$ to one letter, number or word to be come all these nodes (5 node in one cluster) in the same number, letter or word. The next step is to assign all the compressing nodes in the first $K_{2,3}$ to “All” followed by a number with a color is assigned around the compressing node, for example, “All_1” with green color is assigned around the compressing node which meant this node call “All” is mean: B1, B2, B3, B4 and B5. “All_2” with red color is assigned around the compressing node which meant this node call “All” is mean: B6, B7, B8, B9 and B10. On the other hand, these $K_{2,3}$ which are not in one $K_{2,3}$ we will deal with it as normal $K_{2,3}$ (compress) such is: Tomato color is assigned around the compressing node for the fifth $K_{2,3}$ with the “B11.” Pink color is assigned around the compressing node for the fifth $K_{2,3}$ with the "B12." Salmon color is assigned around the compressing node for the fifth $K_{2,3}$ with the "B13." Purple color is assigned around the compressing node for the fifth $K_{2,3}$ with the "B14" as shown in figure 28.

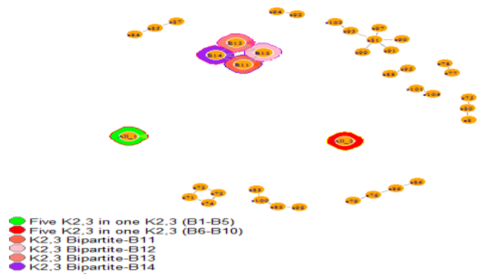

Fig. 28: Tree Layout of 14 $K_{2,3}$ (Second Compress). After compressing the $K_{2,3}$ and creating the new graph,

The next step is to check if there is another $K_{2,3}$ in the graph by using the Linkcomm Package feature again (loop)as shown in figure 29.

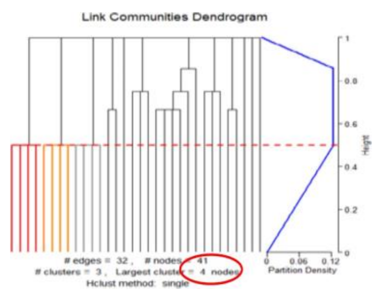

Fig. 29: Linkcomm Package- Second Compress the $K_{2,3}$. In this dataset, there are 14 $K_{2,3}$ in the graph and some of these $K_{2,3}$ in one $K_{2,3}$.

The last part of the algorimetic solutions of the hypothesis was used to "Represent $K_{2,3}$" through R. To represent these $K_{2,3}$ by R, we selected and customized one of the available layout which is Tree Layout as shown in figure 30 and 31.

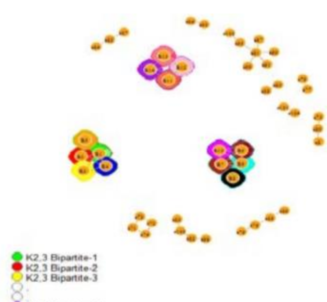

Fig. 30: First Represent Graph by Tree Layout.

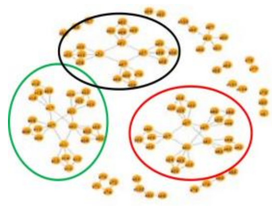

Fig. 31: Second Represent Graph by Tree Layout.

9. DISCUSSION RUSULTS

9.1. Runtime

The average of 100 runtime by R for these datasets show that when the datasets have two levels in $K_{2,3}$ ($K_{2,3}$ in $K_{2,3}$) the average of 100 runtime increases in comparison to small, one-level, datasets. More levels in the datasets increase 100 runtime are to a more complex algorithm (e.g. two level in $K_{2,3}$ in the dataset meant this dataset is a complex dataset) as shown in figure 32 [Heymann & Grand, 2013].

Dataset	n	e	Average of 100 Runtime	Gig-O
Small Datasets 1 (No $K_{2,3}$ Bipartite)	15	12	2.65 secs	2.60 secs
Small Datasets 2 (One $K_{2,3}$ Bipartite)	15	13	2.628 secs	2.62 secs
Small Datasets 3 (Two $K_{2,3}$ Bipartite)	15	17	2.684 secs	2.68 secs
Small Datasets 4 (Five $K_{2,3}$ Bipartite)	25	36	3.04 secs	3.18 secs
Big Datasets 1-AnAge (No $K_{2,3}$ Bipartite)	197	150	3.90 secs	4.83 secs
Big Datasets 2-Media (14 $K_{2,3}$ Bipartite)	124	103	4.401 secs	4.44 secs
Big Datasets 3-Time Use (30 $K_{2,3}$ Bipartite)	238	505	5.639 secs	5.33 secs

Fig 32: Average of 100 Runtime for all the Dataset by R.

9.2. Big-O(BigOh) For Time Complexity Of Algorithms

Big-O [Tardos & Kleinberg, 2006] is an expression for the execution time of a program (how long the program will take). There is theorem that we will use it in this part [Parker & Lewis, 2016]:

- **Theorem**: $O((e+n)n)log(e+n)+log(n)$
- **n** = number of nodes
- **e** = number edges
- **RAM** = 4.00 GB
- **CPU** = 2.40 GHZ
- **System type** = 64-bit

Comparison between the outcomes of the average of 100 runtime coulmen and the Big-O(BigOh) coulmen for the Time Complexity of Algorithms show us that these columns are equal which meant that this theorem is a correct as showen in figure 5 [Joyner, Nguyen & Cohen, 2010]:

Dataset	n	e	Average of 100 Runtime	Gig-O
Small Datasets 1 (No $K_{2,3}$ Bipartite)	15	12	2.65 secs	2.60 secs
Small Datasets 2 (One $K_{2,3}$ Bipartite)	15	13	2.628 secs	2.62 secs
Small Datasets 3 (Two $K_{2,3}$ Bipartite)	15	17	2.684 secs	2.68 secs
Small Datasets 4 (Five $K_{2,3}$ Bipartite)	25	36	3.04 secs	3.18 secs
Big Datasets 1-AnAge (No $K_{2,3}$ Bipartite)	197	150	3.90 secs	4.83 secs
Big Datasets 2-Media (14 $K_{2,3}$ Bipartite)	124	103	4.401 secs	4.44 secs
Big Datasets 3-Time Use (30 $K_{2,3}$ Bipartite)	238	505	5.639 secs	5.33 secs

Fig 33: Big-O(BigOh) for Time Complexity of Algorithms.

10.CONCLUSION

It is a feasible to use $K_{2,3}$ as a key vector to prune a complicated graph. Feasible is means: Dataset is doable (is works) and time is tolerable. The mathematical theorem ($O((e + n)n)$) is useful for the algorithm. Useful means: The average of 100 runtime results were identical to the Big-O results.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my deepest appreciation to The Saudi Arabian Cultural Mission (SACM) for giving me their valuable suggestions and sup- port throughout my research.

REFERENCES

Awal. T, Rahman. M. (2010). A linear algorithm for resource four- partition four-connected planar graphs, 6th International Conference on Electrical and Computer Engineering ICECE. , pp. 526- 529.

Bai. B, Chen. Letaief. W. K, & Cao. Z. (2011). 'Diversity-Multiplexing Tradeoff in OFDMA Systems: An -Matching Approach', IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, vol. 10, pp. 3675- 3687.

Birmingham, UK: Packt Publishing Ltd Etemad. K, Carpendaleya. S & Samavatiz. F. (2014). Spirograph inspired visualization of ecological network, pp. 81-91.

Carter. B. (2008). Network: A package for managing relational data in R, Journal of Statistical Software, vol. 24, pp. 1-36.

Carsten F. Dormann., J. Fruend., B. Gruber., with additional code from Devoto. M, Iriondo. J, Strauss. R, & Vazquez. D, also based on C-code developed by Bluethgen. N, A. Clauset/Rouven Strauss & Rodriguez-Girones. M. (2014). Packagebipartite, pp. 1- 154.

Csardi. G. (2015). Package igraph, Network Analysis and Visualization, pp. 1-462.

Dunne. C & Ben. S. (2013). Motif simplification: improving network visualization readability with fan, connector, and clique glyphs, Changing Perspectives, Paris, France, pp. 3247-3256.

Dormann. C, Bernd. G, & Jochen. F.(2008). Introducing the bipartite package: analysing ecological networks, vol. 8/2, pp. 7-11, 2008.

Egawa. Y & Furuya. M. (2014). Forbidden triples containing a complete graph and a complete bipartite graph of small order,Graphs and Combinatorics, pp. 1149-1162, 2014.

Ghosh, B. R., Banerjee. S, Dey. S, Ganguli. S & Sarkar. S. (2014). Off-Line signature verification system using weighted complete bipartite graph, 2nd International Conference on Business and Information Management (ICBIM), pp. 109-113.

Gori. G, Maggini. M., & Sarti. L. (2005). Exact and approximate graph matching using randomly walks, IEEE TRANSACTIONS ON PAT- TERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 27, pp. 1100-1111.

Gentry. J, Gentleman. R & Huber. W. (2015). How to plot a graph using rgraphviz, pp. 1-25.

Hartung. E. (2006). The linear and cyclic wirelength of complete bipartite graphs.pp. 1-20.

Tardos. E. (2006). and Kleinberg. J, 'Algorithm Design'. Pp. 1- 10.

Heymann, S., Grand, L., & Bndicte. (2013). Visual analysis of complex networks for business intelligence with gephi', 17th International Conference on Information Visualisation, vol. 13, pp. 307-312.

Hu. Y, Jin. H, Lin. S & Li. H. (2010). A study of pseudo-automorphism groupabout the cn',Second International Conference on Informa- tion Technology and Computer Science, vol. 10, pp. 332-335.

Joyner. D, Van Nguyen. M, Cohen. N. (2009-2010). Algorithmic Graph Theory', pp. 1-163.

Kalinka. A. T.(2014). The generation, visualization, and analysis of link communities in arbitrary networks with the R package linkcomm. Vienna, Austria. pp.1-16.

Ken, Cherven. (2015). Mastering Gephi Network Visualization.

Koutra. D. (2015). Exploring and making sense of large graphs. Sub- mitted in partial

fulfillment of the requirements for the degree of Doctor of Philosophy, pp. 1-231.

Katherine. O. (2015-Aug-11). Static and dynamic network visualization WithR. Retrieved from https://rpubs.com/kateto/netviz.

Los Angels, CA. London, UK. New Delhi, India. Washington, DC: MPG Printgroup

Martin, G. Everett., Jeffrey, C. H., Stephen. B. (2013). Analyzing Social Networks.

Maris, R. L. (2008). Statistical Computing with R. Bowling Green, OH: Bowling Green University.

Nocaj. A, Ortmann. M,&Brandes. U. (2015-Sept-11).Untangling hairballs. From 3 to 14 Degrees of Separation. Retrieved from https://www.unikonstanz.de/mmsp/pubsys/publishedFiles/NoOrBr14.pdf.

O'HAIR, K. (2009) 'The geometric structure of spanning trees and appli- cations to multiobjective optimization' pp. 1-41.

Parker. M, Lewis. C. (2016-February-15). Why is Big-O Analysis Hard?. Retrieved from http://dx.doi.org/10.1145/2526968.2526996.

Ryan. A, & Mark. H. (2008) 'Networksis: a package to simulate bipartite graphs with fixed marginals through sequential importance sam- pling', Journal of Statistical Software, vol. 24, pp. 1-21.

Rosen. K. (2012). CS 137 - Graph theory - lectures 4-5, Discrete Mathe- matics and its Applications, 5th ed, chapters. 8.7, 8.8, 2012.

Richard Brath, David Jonker. (2015). Graph Analysis and Visual- ization. Indianapolis. IN: John Wiley & Sons, Inc. S. Lehmann. S., Schwartz. M & Hansen3. K. (2013). Bi-clique Communi- ties, 1Center for Complex Network Research and Department of Physics', Northeastern University, Boston, pp. 1-10.

Solomon. M. (2015-Aug- 11). Working with bipartite/affiliation network data in R. Generating Labels for Supervised Text Classification using on plotting numeric data by groups. Retrieved from https://solomonmessing.wordpress.com/2012/09/30/working-with-bipartiteaffiliation-network-data-in-r/.

(2016-January-1).The Open Graph Viz Platform. Retrieved from https://gephi.org/.