Number of MSTs in weighted K_n: Some configurations and trends

Parth Rathod-201401089
DA-IICT,Gandhinagar
Mentor : Prof. Rahul Muthu
April 26, 2018

***Abstract* - This is an extension of BTP projects done by students of 2007 and 2010. The project is about studying trends in the number of minimum spanning trees in weighted complete graphs by varying the edge weight assignments in a controlled manner. Specifically, we edge weighted complete graphs where all edges are of weight 1 or 2. We focus on 7 specific subgraphs and study the number of MSTs when edges of these subgraphs are of weight 1 and remaining edges are of weight 2. These subgraphs being acyclic, all the edges of weight 1 are present in each MST. Subsequently we study the reversed scenario where the edges of these 7 subgraphs are of weight 2 and rest of the edges are of weight 1. We also tried to understand the different trends observed after finding total MSTs. So we wrote a code in Java and used a MATLAB tool in order to find the total number of Minimum Spanning trees for the various configurations of the Graph.**

Keywords - Minimum Spanning trees, Weight Class, Connected Components, Forced Edges and Forbidden edges.

1 Introduction

The spanning trees definition can be understood by the name itself. Spanning means that it spans all the vertices and a tree meaning it is connected and has no cycle. So, a Minimum spanning tree is a spanning tree with lowest weight among all the possible trees. A graph can have several Minimum Spanning Trees.

At the beginning the project was about figuring whether it is possible to assign edge weights to the complete graph K_n resulting in exactly k MSTs, for a given pair of positive integers n, k. Where $1 <= k <= n^{(n-2)}$. Given the magnitude and the difficulty of this problem we chose to focus on specific simple aspect of this problem.

We consider 7 specific configurations (all acyclic) with number of edges varying from 1 to 3 and study the general problem under these scenarios:
A) Edges of these configurations have weight 1 and rest of the edges have weight 2.
B) Edges of these configurations have weight 2 and rest of the edges have weight 1.
The number of vertices of the complete graphs considered is at least one more then the number of vertices in the given configurations. In scenario A edges of these configurations are going to be present in all MSTs. Whereas in scenario B edges of this configurations are going to be absent from all MSTs. Cayley's formula is also used in order to find the total possible spanning trees. Explanation and the Examples for the same are given in the rest of the sections of the report.

The seven configuration we consider are:

- P_2 - Path of length 1.

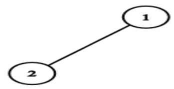

Figure. Graph of P_2

- M_2 - Matching of 2 edges. Where matching means that no edge shares the common vertex.

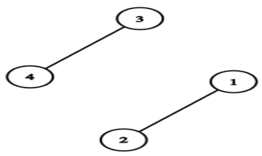

Figure. Graph of M_2

- P_3 - Path of length of 2.

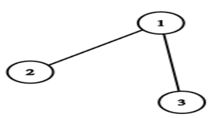

Figure. Graph of P_3

- M_3 - Matching of three edges.

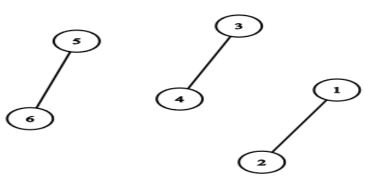

Figure. Graph of M_3

- $P_3 + M_1$ - Path of length 2 and a edge which does not share a common vertex with P2.

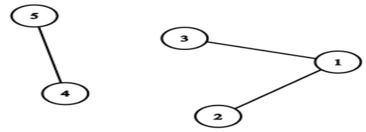

Figure. Graph of $P_3 + M_1$

- P_4 - Path of length 3.

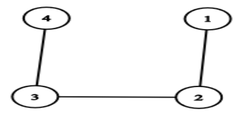

Figure. Graph of P_4

- $K_{1,3}$ - 3 edges from 3 different vertex adjacent to single vertex.

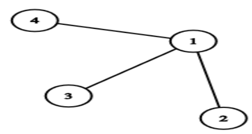

Figure. Graph of $K_{1,3}$

Definition of Forced and Forbidden components

All the configurations listed above are acyclic. Thus when we give the edges of those configurations weight 1 and rest of the edges weight 2, the edges of the configurations are present in all MSTs and are thus called **forced**. Since the number of vertices of the complete graph is assumed to be at least one more then the number of vertices of the configurations, the edges of the configuration do not disconnect the graph. Consequently, when we assign the edges of the configurations weight 2 and rest of the edges weight 1, the edges of the configuration are absent from all MSTs and are thus **forbidden**. We illustrate this concept with an example. Definition of forced subgraph is such that if P_3 is a forced subgraph in the complete graph of n=5 vertices. So there is going to be (5*(4))/2=10 edges. Then 2 adjacent edges which creates a path of length of 2,each edge will have edge weight 1 and rest of the 8 edges will have edge weight 2. Which means that in all MSTs this subgraph is always going to be present.

2 Motivation and Predictions

Based on graph theoretic concepts we had some intuition on the relative values of number of MSTs under certain changes to number of vertices as well as weight assignment. It was difficult to confirm our intuition with precise mathematical proofs and so we chose to verify our intuition empirically by writing code and studying test cases. We list here some potential results based on our intuition. These are empirically confirmed in later sections.

1. For a fixed configuration, as n increases asymptotically the number of MSTs in the forced case is significantly smaller than the number of MSTs in forbidden case.
2. For forced configurations, as the number of edges increases the number of MSTs decreases.
3. For forbidden configurations, also as the number of edges increases the number of MSTs decreases.
4. For two configurations with the same number of edges the number of MSTs is the same if the tree-wise number of vertices is identical.
5. For two configurations, with same number of edges, the one with more components results in more MSTs.
6. When one configuration has fewer edges but only one component and the other configuration has more edges and more components, the trends (2) and (5) are working against each other. This might result in an anomaly.

However for the configurations and values of n for which we were able to generate outputs the trend (6) did not manifest itself and hence could not be empirically verified. The nature of code we wrote is quite complex and requires customization for each configurations and the complexity explodes with the size of the configuration. Thus it was infeasible to test our predicted trend (6), and this could possibly be done in future.

3 Approach and Methodology

I created a Java program and also used MATLAB tool for solving my problem. I worked on seven different subgraph for my Project. We describe our approach for the forced and forbidden cases separately in the following two subsections.

3.1 Edges of the configurations are Forced

For solving this problem of counting total MSTs when edges of the configurations are forced, the basic idea is to count the number of ways to connect the forced subgraph with all possible forests of the remaining vertices. Also while connecting forced subgraph with forest of rest of the vertices following points were used to count MSTs efficiently and correctly:

1. Find all possible ways to connect a forest with forced subgraph, such that it becomes a single connected component.
2. Edges are to be added such that there are exactly $n-1$ edges.
3. If the forced subgraph have more than one component, then we have to find all possible ways to connect its component with each other. Then connect it with rest of the forest. For example, there will be 3 ways to connect M_3 within itself. First when all 3 edges are connected.

Second when 2 edges are connected and third edge is not connected. Third when none of the three edges are connected with each other.

In order to find all the possible forests, we wrote a recursive code which finds all the possible ways to partition all the vertices other than forced subgraph. For example, if there are 3 vertices other than forced subgraph then there are following ways to create forests and each of it will be connected with forced subgraph. Each following ways represent the order of the each of the tree in their forest:

- $1, 1, 1$
- $1, 2$
- 3

Example for P_3 forced in K_9

Let's take an example of K_9 and P_3 is a forced subgraph in it. There are 6 vertices other than the forced subgraph. We considers that edges $[1, 2]$ and $[2, 3]$ will create P_3. The $[1, 2], [2, 3]$ edges are going to be present in all the possible MSTs since it is a component with lower weight. As we discussed earlier each tree of the forest will be connected with the forced subgraph, similarly we will find number of ways to connect all possible forests with forced subgraph. For our current example, let a forest be $1, 2, 3$. So the forest have 3 trees of order 1,2 and 3. There are following number of ways to connect forest with forced subgraph in our example:

MSTs= $\binom{6}{3}*\binom{3}{2}*3^{(3-2)}*2^{(2-2)}*3*3*2*3*1*3$

where, $\binom{6}{3}$= number of ways to select 3 vertex out of 6
$\binom{3}{2}$= number of ways to select 2 vertex out of remaining 3 vertex
$3^{(3-2)}*2^{(2-2)}$=number of ways to create trees from selected vertices using Cayley's formula
3*3*2*3*1*3= number of ways to connect each of this component with P_3.
Basically to connect each tree of forest with forced subgraph(which is a tree in this case). We find the total number of ways to connect both trees with one edge. Since this trees are subgraph of complete graph and vertices of each tree would be adjacent with each other in original graph. There would be following number of ways to connect both trees with one edge:

- no. of ways=$|P_3|$* order of tree in the forest

Similarly code will find the total MSTs for all the possible forest.

Complexity of code increasing with increase in number of component in forced configurations

In the case M_3, none of the three edges shares a common vertex. There are 3 ways to connect this Matching of 3 edges as already explained. When none of the edges are connected there are following ways to connect this forced configuration with rest of the forest. Let say 1 to 6 vertices will create M_3. So forced configurations have edges [1,2], [3,4], [5,6].

- Connect 2 edges with one tree of the forest and let this connected component be name $'A'$. Now connecting third edge of matching with separate tree of forest and then connecting that tree with forced configuration from $'A'$. Then we can connect rest of the trees from the forest with this single connected component

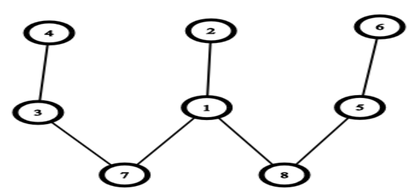

Figure. Connecting M_3 via 2 trees of forest

- Similarly we can connect three edges to one tree of forest and then connecting rest of the tree from forest with this single connected component.

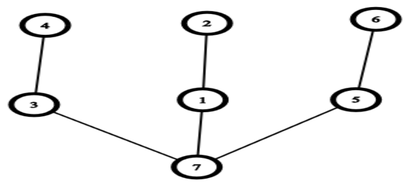

Figure. Connecting M_3 via 1 tree of forest

So with increasing number of forced subgraphs the complexity of the code will also increase. Since the number of ways to connect within the forced subgraph and then number of ways to connect with the rest of the graph will also increase.

3.2 Edges of the configurations are Forbidden

For solving this problem of counting total MSTs when edges of the configurations are forbidden, all the possible MSTs are counted after removing the edges of forbidden configuration. All other edges are of weight 1, so we can consider it as simple unweighted graph. Since all the edges of weight 2 are removed. Therefore we used a MATLAB tool named MATGRAPH, which counts the total number of spanning trees for simple unweighted graph.

4 Observations

This section is going to focus on the different trends observed when different configurations are either forced or forbidden. We created Bar Graphs and Line Graphs of the data observed from the Java code and MATLAB tool in order to better understand the different trends.

Forced Components in K_7

When there are forced configurations in the graph K_7. As already discussed forced configurations will be present in all MSTs.

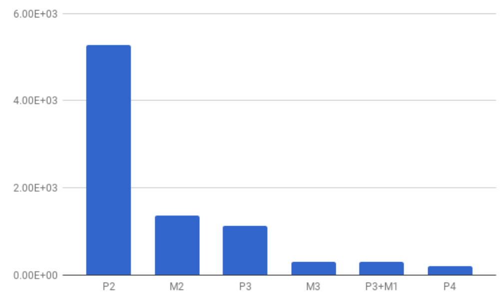

Figure. Comparisons of no. of MSTs when there are forced components for graph with n=7

Following trends are observed with the help of a bar graph.

1. It is clearly visible that it results in more number of total MSTs when forced components are of comparatively of smaller size.

2. As we increase the size of forced components the number of MSTs decreases gradually.

3. When only one edge is forced in the graph for $n = 7$ then number of MSTs are 5282, whereas just forcing one more edge and making the forced component as P_3 the number of MSTs decreases to 1137. Which is a really a large decrease in the number of MSTs.

4. Also there are few other interesting observations such as comparison between P_3 and M_2 as well as P_4 and M_3. That there will be more MSTs in case of M_2 as compared to P_3 even though M_2 has two component and even though M_2 is affecting more number of vertices then P_3. In case of M_2 there are 1372 possible MSTs as compare to MP_3 which gives 1137 number of MSTs. Similar trend is observed in case of P_4 and M_3. Where there are 296 possible MSTs for M_3 as compare to 196 possible MSTs for P_4.

Forced Components in K_10

Now we are forcing different configurations in complete graph with n=10 vertices and observing trends.

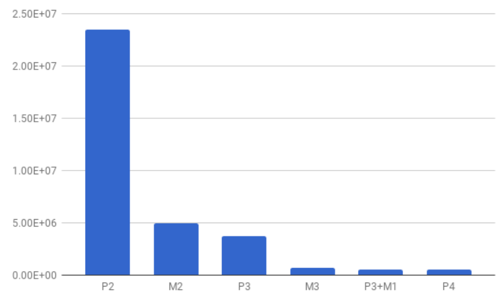

Figure. Comparisons of no. of MSTs when there are forced components for graph with n=10

Similar trends can be observed in this graph also as compare to the complete graph with n=7. The only difference is that as we increase the value of the n, the value of MSTs decreases more gradually for larger forced components as compare to fall for smaller values of n.

Forbidden Components in K_7

Now we have observed the following trends when there are Forbidden Components in K_7

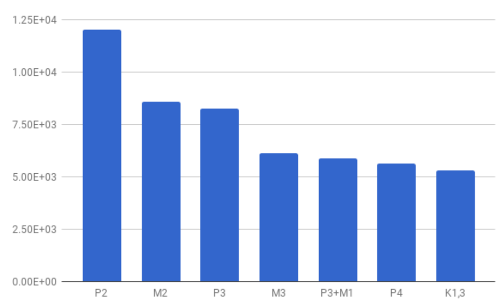

Figure. Comparisons of no. of MSTs when there are forbidden components for graph with n=7

1. As it can be clearly seen from the graph that as we forbid bigger components the decrease in the number of MSTs is relatively less gradual as compare to the trees when there were forced components.

2. Whereas there is gradual decrease if we compare complete graphs of P1 and P2 as forbidden component. The number of MSTs for P1 as forbidden component is 12005 whereas for P2 as forbidden components there are 8232 MSTs only. Which is a significant fall with the forbidding just one more edge as compare to P1.

3. In the case of Forbidden configuration also M_2 forbidden in K_7 results in more number of MSTs as compared to when P_3 is forbidden in K_7. Similar observations are made for all the values of n till 15.

Forbidden Components in K_10 Now we have observed the following trends when there are Forbidden Components in K_10:

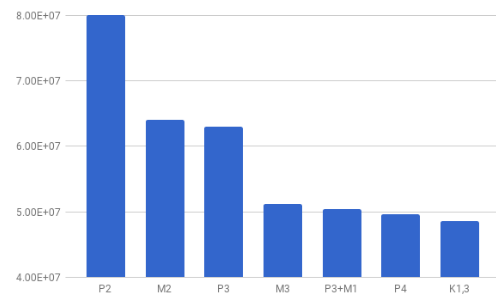

Figure. Comparisons of no. of MSTs when there are forbidden components for graph with n=10

Similar trends are observed in case of K_10 as compare to K_7. Only difference is that, as it can be clearly seen from the plotted graph that decrease in number of MSTs is more gradual for different forbidden configurations as we increase the number of vertices.

Comparison between MSTs for Forced and Forbidden components We have compared the number of MSTs obtained for different values of n from 1 to 15, when M_3 is forced is compared to when M_3 is forbidden and we have observed the following trends from the graph:

1. As it can be clearly seen that from beginning itself there will be more MSTs when M_3 is forbidden as compare to when M_3 is forced.

2. As the value of n increases the difference between number of MSTs increases. For example, when $n = 15$. If M_3 is forced

then there will be 6.87085E+12 MSTs, whereas if it M_3 is forbidden there will be 1.27E+15 MSTs.

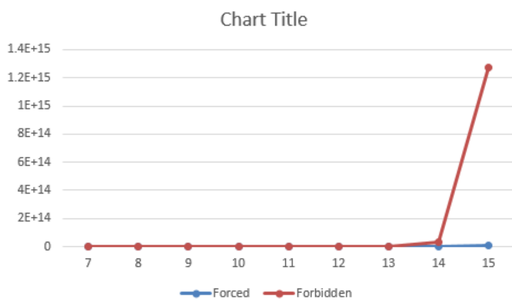

Figure. Comparisons of no. of MSTs when M_3 is forced with when its forbidden (x-axis is number of vertices, y-axis is number of MSTs)

Graph presentation of difference in number of MSTs when different configurations are forced

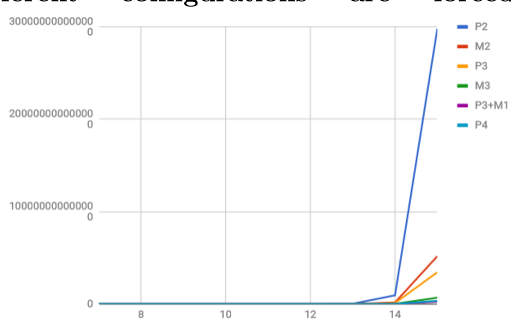

Figure. Line Graph for number of MSTs for increasing number of vertices for forced configurations (x-axis is number of vertices, y-axis is number of MSTs)

As we can clearly see that as the number of vertices increases. The number of MSTs for forced component with smaller size increases more exponentially as compared to MSTs with forced component of large size.

Graph presentation of difference in number of MSTs when different configurations are forbidden

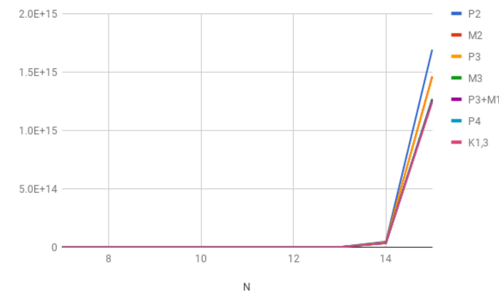

Figure. Line Graph for number of MSTs for increasing number of vertices for forbidden configurations (x-axis is number of vertices, y-axis is number of MSTs)

As it can be seen in this line graph also that as the size of forbidden configuration increases the number of MSTs decreases.

Following are the data found from the Java code when 7 chosen configurations are forced

N	P2	M2	P3	M3	P3+M1	P4 or K1,3
3	2	NA	NA	NA	NA	NA
4	8	NA	3	NA	NA	NA
5	50	20	15	NA	NA	4
6	480	144	108	NA	36	24
7	5282	1372	1137	296	294	196
8	76456	17920	13908	3136	2964	2240
9	1219122	270756	215217	41736	38286	30084
10	23422720	4974400	3697410	717824	563400	497440
11	4.97E+08	1.01E+08	74539113	13946312	9841146	9196324
12	1.21E+10	2.35E+09	1.66E+09	3.19E+08	1.9E+08	1.96E+08
13	3.18E+11	6E+10	4.18E+10	8.01E+09	4.19E+09	4.61E+09
14	9.42E+12	1.7E+12	1.14E+12	2.25E+11	1.01E+11	1.21E+11
15	2.98E+14	5.18E+13	3.44E+13	6.87E+12	2.72E+12	3.45E+12

Following are the data found from the MATLAB tool(Matgraph) when 7 chosen configurations are forbidden

N	P2	M2	P3	M3	P3+M1	P4	K1,3
3	1	NA	NA	NA	NA	NA	NA
4	8	4	3	NA	NA	NA	NA
5	75	45	40	NA	NA	21	16
6	8.64E+02	576	540	NA	360	336	300
7	1.20E+04	8575	8.23E+03	6125	5.88E+03	5.64E+03	5.29E+03
8	1.97E+05	1.47E+05	1.43E+05	1.11E+05	1.08E+05	1.04E+05	100352
9	3.72E+06	2.89E+06	2.83E+06	2.25E+06	2.20E+06	2.16E+06	2.10E+06
10	8.00E+07	6.40E+07	6.30E+07	5.12E+07	50400000	4.96E+07	4.86E+07
11	1.93E+09	1.58E+09	1.56E+09	1.29E+09	1.28E+09	1.26E+09	1.24E+09
12	5.16E+10	4.30E+10	4.26E+10	3.58E+10	3.55E+10	3.51E+10	3.47E+10
13	1.52E+12	1.28E+12	1.27E+12	1.09E+12	1.08E+12	1.07E+12	1.06E+12
14	4.86E+13	4.17E+13	4.14E+13	3.57E+13	3.55E+13	3.52E+13	3.49E+13
15	1.69E+15	1.46E+15	1.45E+15	1.27E+15	1.26E+15	1.25E+15	1.24E+15

5 Conclusion

I would like to conclude this report by saying that I learned some new concepts related to graph theory, and got familiar with the problems which i was not aware of and tried to solve it using programming and other MATLAB tool(MATGRAPH). From the above observations I can conclude that as we force components of bigger size in the graph, it reduces the possible number of MSTs and this trend gets more clearer as we increase the value of n. Whereas when we forbid certain components and as we increase size of forbidden components decrease in value of MSTs is not very gradual as forced configurations. But in this case also when we increase value of n, it will represent gradual decrease in the values of MSTs.

Acknowledgement

I would like to express my gratitude towards Prof. Rahul Muthu providing me with the opportunity and helping me throughout my BTP project.

6 References

- Matrix Tree Theorem
  https://en.wikipedia.org/wiki/Kirchhofftheorem
- Cayley's formula
  https://en.wikipedia.org/wiki/Cayleyformula
- MATLAB tool *Matgraph*
- Complete Graph
  https://en.wikipedia.org/wiki/Complete$_g$*raph*