Wireless Sensor Network

Final Report

Presented by
Luke Roosje
Jessica Denney
Zuli Wu

Submitted to the
Department of Computer Science
Prof. Jing Deng
University of North Carolina at Greensboro

11.29.2017

Contents

1 Introduction **2**
1.1 Question Description . . . 2

2 System Model **3**
2.1 Basic Assumptions . . . 3
2.2 Flow Diagram . . . 3

3 Approach and Program Modules **6**
3.1 Algorithm Description . . . 6
3.2 Penalty Parameter . . . 6
3.3 Algorithm Weaknesses . . . 7

4 Results and Discussions **8**
4.1 Limiter Results . . . 8
4.2 Cost Evaluation . . . 9
4.3 Cost Evaluation for Individual Runs . . . 9
4.3.1 Cost Evaluation for Run 1 . . . 9
4.3.2 Cost Evaluation for Run 2 . . . 11
4.3.3 Average Cost Comparison for Run 1 and Run 2 . . . 12
4.4 Data Trend with Limiter 3 . . . 13
4.5 Data Trend with Limiter 5 . . . 13
4.6 Data Trend with Limiter 10 . . . 14
4.7 Data Trend with Limiter 15 . . . 14

5 Conclusion **15**
5.1 Summary of Project . . . 15

Chapter 1

Introduction

1.1 Question Description

In this project, we work on randomly distributed sensor networks and try to find the most efficient way to collect data with MATLAB. There are 1000 sensor nodes in a region of 1,000x1,000 m^2. We have to put 4 data sinks and find the shortest path (in terms of hops) to the nearest data sink for each node.Besides,as for those nodes which cannot reach any data sink, we assign a penalty (p) as cost value. Additionally, we experiment with different value of penalty to check its influence of result.Our final goal is to find the proper position for 4 data sinks with optimized cost value.

Chapter 2

System Model

2.1 Basic Assumptions

Sensor nodes are randomly distributed so there is no certain pattern here. According to the question description, we have the following idea. If we put data sinks in the area where there are lots of sensor nodes nearby and if we have few disconnected nodes, the final cost may be close to optimized cost. Therefore,we plan to divide the whole region into different sectors and calculate nodes density. Based on this, we design our algorithm.

2.2 Flow Diagram

We design a data flow diagram for better understanding of the process for project. It has flowing 8 steps:

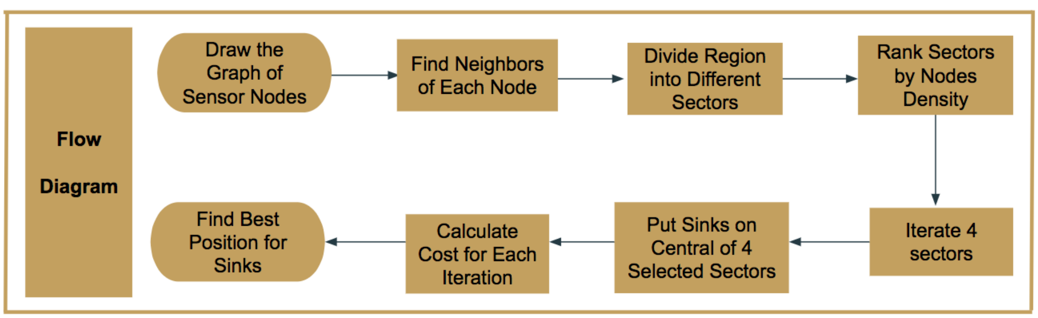


Figure 2.1: Flow Diagram

1. We randomly distribute 1000 sensor nodes in a region of 1000*1000 m ^{2}.

2. For each node, we find all its neighbors and draw solid line between them. By the way, neighbors refer to those nodes whose distance is smaller than 50 m.

3. We set a variable called limiter, which is integer. We divide the region into different sectors using limiter as parameter. For example, if the limiter = 2, the area is divided into 2*2 sectors equally.If the limiter = 3, the area is divided into 3*3 sectors equally.If the limiter = s, the area is divided into s*s sectors equally. The process goes as the following graph:

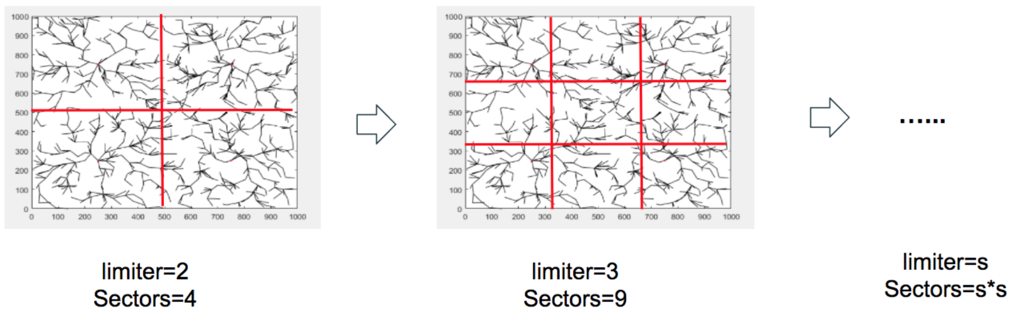


Figure 2.2: Divide Region

4. We calculate the node density for each sector by counting nodes.Then rank sectors by nodes density.

5. Iterate 4 sectors from lowest density to high density. If sectors number equals 4 (when limiter = 2), there will be only one iteration 4,3,2,1. If sector number equals 9 (when limiter = 3), there will be 6 iterations. The first iteration is 9,8,7,6,the second iteration is 8,7,6,5,the third iteration is 7,6,5,4, the forth iteration is 6,5,4,3, the fifth iteration is 5,4,3,2 and the last iteration is 4,3,2,1. In conclusion, if sectors number equal s*s (when limiter = s), there will be (s*s-3) iterations from the lowest 4 sectors to the highest 4 sectors.

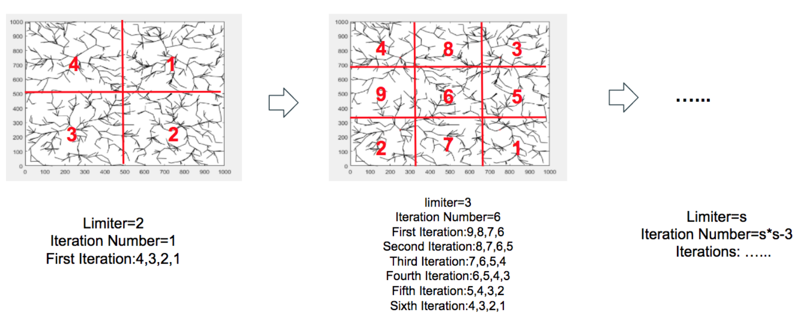


Figure 2.3: Rank and Iterate Sectors

6. For each iteration, we put 4 data sinks on the central of each selected sector.

7. We calculate total cost by adding up the cost for connected nodes and penalty for all disconnected nodes .The cost for connected nodes equals the hop number from each node to its nearest data sink.

8. By comparing the cost value of each iteration, we find the best position for data sink with minimal total cost value.

Chapter 3

Approach and Program Modules

3.1 Algorithm Description

The Matlab code is relatively straightforward, and there is no class or method here. For easier to understand,We explain it with Pseudo code under the method of"DataCollection".[1]

```
 method DataCollection
        DefineSensorNodes
        DefineProcessedNodes
        DefineCriticalVariables
        FindAllNeighbors
        DivideRegionIntoGrids
        FindNodeDensityForSectors
        RankSectors
        ItrateEvery4Sectors
        PlaceDataSinks
        RedeclareNeighborsOfNewlyPlacedSinkNodes
        FindPathFromNodesToSink
        AddUpCostForConnectedNodes
        AddPenaltyForDisconnected Nodes
        AddUpAllCost
        FindBestPosition
end method DataCollection
```

Listing 3.1: *Data Collection*

3.2 Penalty Parameter

We have use different penalty for disconnected node,for example,we set it as 10 and also set it as 20. Since we have very few disconnected node, it doesn't have too much influence on final result.

3.3 Algorithm Weaknesses

Several possible design weaknesses are discovered that directly affect the implementation of this algorithm.

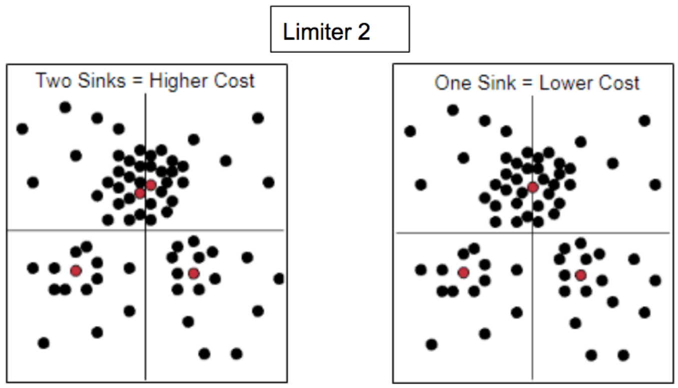


Figure 3.1: Algorithm Weakness

Potential weaknesses within the algorithm arise if high density clusters are spread over more than one quadrant. The resulting cost of putting two sinks in to account for a single cluster is much higher than the alternative of putting a single sink into the center of the cluster.

Additionally, if we have many disconnected nodes, the algorithm will account for the disconnection by applying more weight to those nodes. A potential weakness within arises with the algorithm and random placement of nodes when despite numerous runs there were never any disconnected nodes discovered.

Chapter 4

Results and Discussions

4.1 Limiter Results

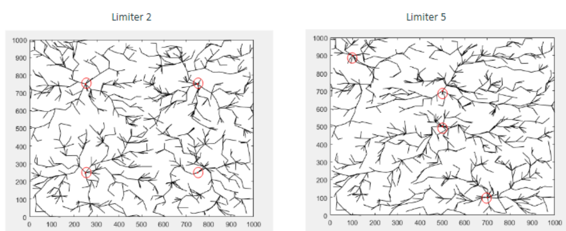


Figure 4.1: Results for Limiter=2 and Limiter=5

The placement of sinks within the highest density quadrants for limiter 2 and limiter 5 can be found above. The sinks for limiter 2 fall onto 250 and 750. The sinks for limiter 5 fall on to the odd numbers of 100, 500, and 700.

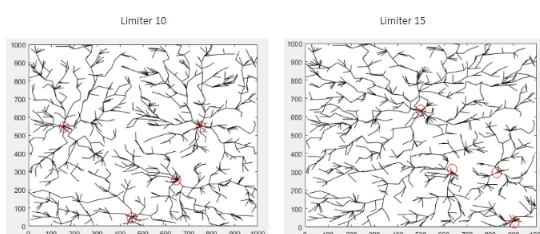


Figure 4.2: Results for Limiter=10 and Limiter=15

The placement of sinks within the highest density quadrants for limiter 10 and limiter 15 can be found above. The sinks for limiter 10 continue the trend from limiter 2 and fall onto 150, 450, 650, and 750. The sinks for limiter 15 fall on to slightly past 500 and the final sink falls onto 900.

4.2 Cost Evaluation

For the best cost efficiency it was found that placing the sinks into the center of the quadrant was necessary; it is important to note however, that the center placement is completely arbitrary and based entirely on the location of nodes and the density of the node cluster. Notably, within a Limiter 5 cluster the sinks will fall into the center of the quadrant, but that does not necessarily apply to all limiters.

The cost in the highest density clusters was found to be 1.7 times lower than when in low density clusters leading to the conclusion that higher densities allow for lower costs. Every node is never more than 7 hops away from the sink, with the furthest distance of seven hops representing the worst case scenario for any node.

The algorithm implemented took into account disconnected node and provided weighted penalties to account for them.

4.3 Cost Evaluation for Individual Runs

4.3.1 Cost Evaluation for Run 1

Numerous runs were conducted in order to test the algorithm, but for analysis purposes only the data from two runs was collected and analyzed. In this particular case the information for Run 1 was compiled into a table and analyzed.

CostRun1 **Limiter**	Highest Density Quadrant Cost	Lowest Density Quadrant Cost	Minimum Cost for the Limiter	Median Cost for the Limiter	Maximum Cost for the Limiter	Mean Cost for the Limiter	Total Cost of the Run
Limiter 2	5964	5964	5964	5964	5964	5964	5964
Limiter 3	7013	7489	7013	7612	8026	7620	45720
Limiter 4	6480	7881	6480	7395	8694	7441	96729
Limiter 5	9776	8664	6305	7968	10032	8058	177284
Limiter 6	7698	9172	6399	8424	12534	8494	280300
Limiter 7	8074	10826	6604	8427	12599	8524	392125
Limiter 8	8474	7773	7038	8605	12376	8686	529840
Limiter 9	8880	10922	6582	8670	14206	9102	709948
Limiter 10	7330	9167	6807	8923	14407	9235	895785
Limiter 11	7909	11091	7232	9500	14672	9732	1148317
Limiter 12	8738	8364	7389	9568	15573	9924	1399333
Limiter 13	13262	9364	6965	10102	14591	10317	1712662
Limiter 14	9578	9468	7083	10086	15864	10387	2004780
Limiter 15	9368	9783	7174	10336	15328	10741	2384452

Several observations can be made regarding Cost Run 1, the first being that the Maximum Cost column provides evidence that the higher the limiter, the more opportunities there are for high density quadrants resulting in higher costs. This trend becomes apparent when one observes the way in which both Maximum Cost and Mean Cost increase throughout the table and end with the Highest Total Cost.

The Highest Density Quadrant Cost reflects the approximate lowest cost for each limiter. This trend is especially noticeable in lower limiters that do not provide very many opportunities for higher density quadrants. In the lower limiters the Highest Density Quadrant Cost always matches the Minimum Cost for the Limiter.

The data for Cost Run 1 presents an interesting outlier with Limiter 13. The Highest Density Quadrant Cost for Limiter 13 is found to be 13,262 causing it to be almost 4,000 more than the cost for the Highest Density Quadrant Cost for Limiter 15 and 6297 more than the Minimum Cost for the Limiter 13.

4.3.2 Cost Evaluation for Run 2

CostRun2 **Limiter**	Highest Density Quadrant Cost	Lowest Density Quadrant Cost	Minimum Cost for the Limiter	Median Cost for the Limiter	Maximum Cost for the Limiter	Mean Cost for the Limiter	Total Cost of the Run
Limiter 2	6206	6206	6206	6206	6206	6206	6206
Limiter 3	7370	6978	6963	7234	8689	7411	44468
Limiter 4	7076	7470	7021	7587	8804	7652	99473
Limiter 5	7437	7251	6455	8139	9980	7987	175711
Limiter 6	7724	6863	6575	7724	9786	7921	261386
Limiter 7	8861	6931	6348	8108	11982	8201	377228
Limiter 8	7593	7127	6424	8220	10354	8375	510857
Limiter 9	8851	7280	6525	8250	13024	8509	663675
Limiter 10	8901	6688	6581	9058	13902	9182	890684
Limiter 11	7550	7405	6961	9296	15351	9647	1138297
Limiter 12	10284	8012	7401	9334	13856	9732	1372191
Limiter 13	7854	8276	7074	9518	13587	9801	1626908
Limiter 14	7550	7692	6427	9771	15376	10007	1931432
Limiter 15	9906	8420	6851	9983	15830	10408	2310475

Several observations can be made regarding Cost Run 2, the first being it continues to provide evidence that was first seen in Cost Run 1 that the Maximum Cost column provides evidence that the higher the limiter, the more opportunities there are for high density quadrants resulting in higher costs. The trend continues that both Maximum Cost and Mean Cost increase throughout the table and ends with the Highest Total Cost.

The Highest Density Quadrant Cost continues to reflects the approximate lowest cost for each limiter.

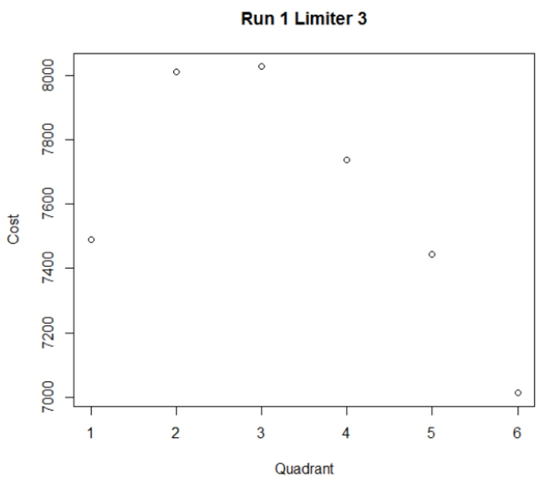
The data for Cost Run 2 presents an interesting outlier with Limiter 12. The Highest Density Quadrant Cost for Limiter 12 is found to be 10,284 causing it to be almost 400 more than the cost for the Highest Density Quadrant Cost for Limiter 15.

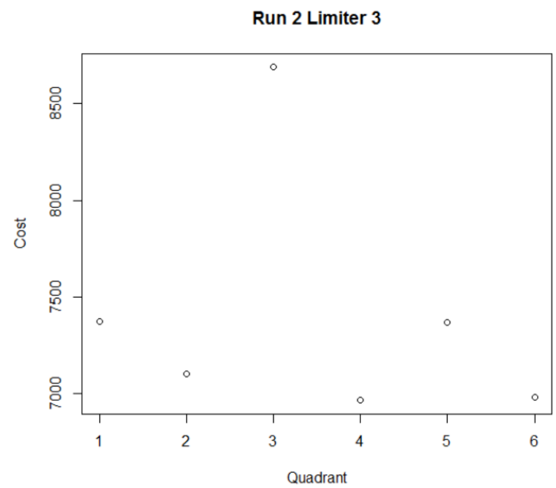
4.3.3 Average Cost Comparison for Run 1 and Run 2

CostRun1 and CostRun2 Average Limiter	Highest Density Quadrant Cost for Run 1	Highest Density Quadrant Cost for Run 2	Average Minimum Cost for the Limiter	Average Median Cost for the Limiter	Average Maximum Cost for the Limiter	Average Mean Cost for the Limiter	Average Total Cost
Limiter 2	5964	6206	6085	6085	6085	6085	9067
Limiter 3	7013	7370	6996	7418	8358	7516	45094
Limiter 4	6480	7076	6975	7515	7923	7546	98101
Limiter 5	9776	7437	6898	7958	9076	8023	176498
Limiter 6	7698	7724	6658	8208	12816	8207	270843
Limiter 7	8074	8861	6892	8243	10756	8363	384677
Limiter 8	8474	7593	7148	8528	10876	8530	520344
Limiter 9	8880	8851	6912	8628	10982	8805	686812
Limiter 10	7330	8901	6968	9152	12392	9209	893235
Limiter 11	7909	7550	7446	9540	13150	9689	1143307
Limiter 12	8738	10284	7906	9504	13948	9828	1385762
Limiter 13	13262	7854	7167	9888	13090	10059	1669785
Limiter 14	9578	7550	6755	10154	14488	10197	1968106
Limiter 15	9368	9906	7321	10346	14368	10574	2347464

The average data for Run 1 and Run 2, found in the table above, shows a distinct trend with increasing costs for higher limiters, but a continuation of the Highest Density Quadrant Costs containing the approximate lowest costs.

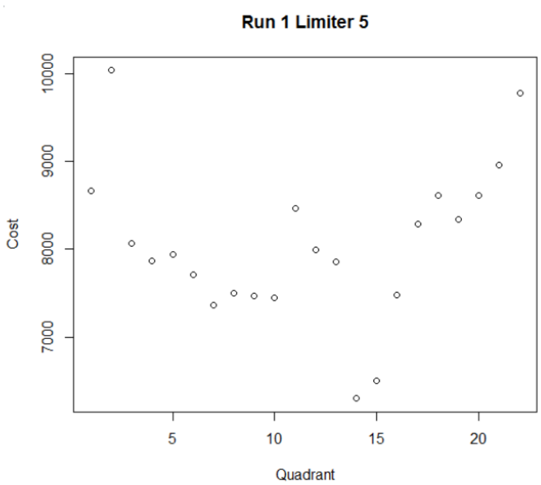
4.4 Data Trend with Limiter 3

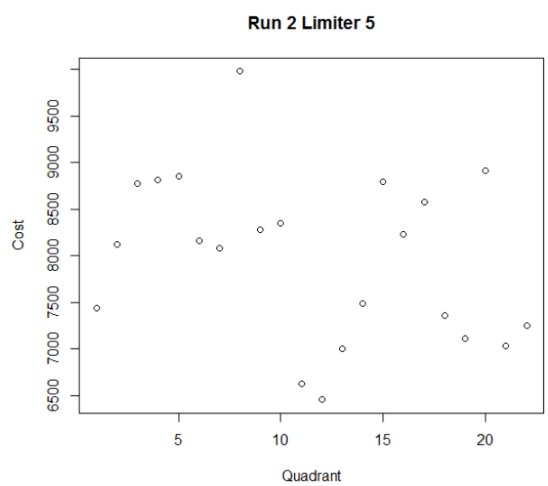




The side-by-side scatterplot graphs for Limiter 3, found above, show the beginning of a trend in which there is a spike in the costs that eventually end with the highest density quadrant having the lowest cost.

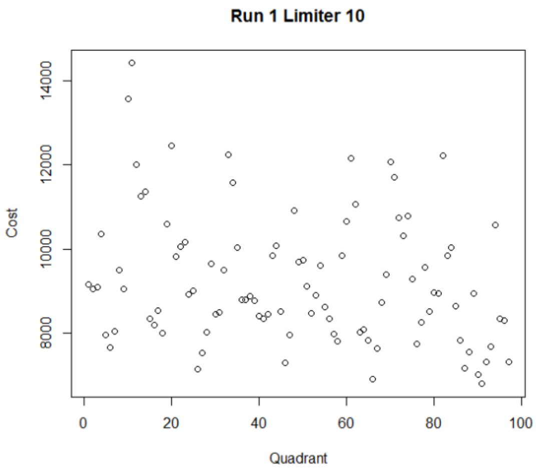
4.5 Data Trend with Limiter 5

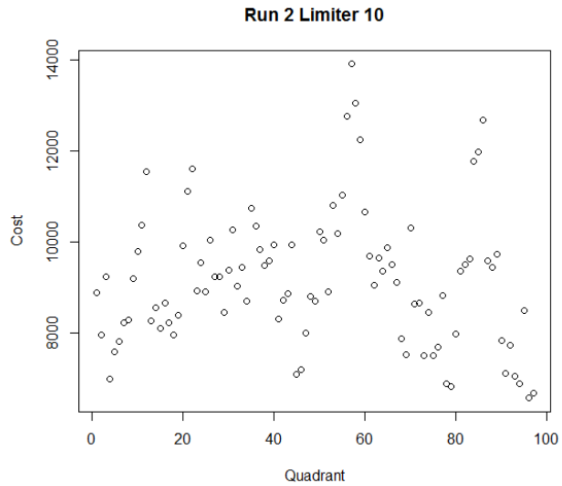




The side-by-side scatterplot graphs for Limiter 5, found above, show the distinct emergence of a trend in which there is a spike in the costs followed by a decrease and another spike before eventually the highest density quadrant has the approximate lowest cost.

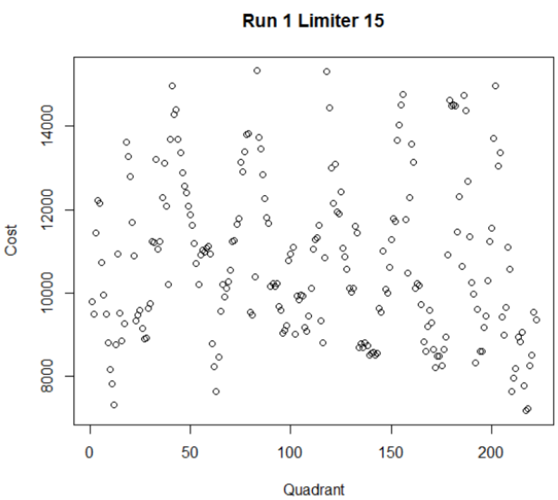
4.6 Data Trend with Limiter 10

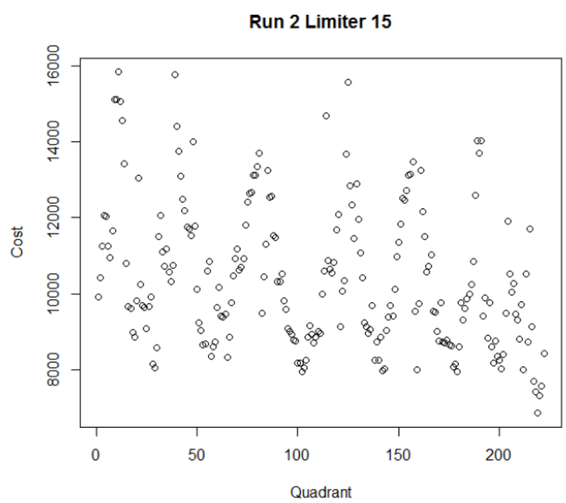




The side-by-side scatterplot graphs for Limiter 10, found above, show a wide range of increases and decreases in cost before tapering off to the lowest costs in the highest density sections.

4.7 Data Trend with Limiter 15





The side-by-side scatterplot graphs for Limiter 15, found above, show a very distinct trend of valleys and peaks that result in the lowest costs in the highest density sections at the end of the graphs.

Chapter 5

Conclusion

5.1 Summary of Project

After numerous runs and extensive quantitative analysis of cost values, we found that the cost in the highest density clusters is 1.7 times lower than when in low density clusters which confirms our previous assumption that ”Higher Density = Lower Cost”. Based on the analysis in which we placed nodes, then arranged the quadrants by density before placing sinks and calculating costs, we have proven our algorithm to be effective.

Additionally, we found that there will be very few disconnected nodes so the penalty value for disconnected nodes is a valid precaution, but it is not incredibly important. Last but not least, we found that seven hops from the sink would be the worst case scenario using this algorithm and that the placement of sinks in the center of the quadrant is the ideal placement in order to achieve the most efficiency and the lowest costs. The presence of outliers within the data represent an unexplained phenomenon within the data. We further concluded that by organizing and placing the sinks in the lowest to highest density quadrants we would be able to finish with lower costs.

Bibliography

[1] Pro. Jing Deng. *Solution of Homework 3*. CSC 568-01, Fall 2007.