FINDING MAXIMAL LOCALIZABLE REGION IN WIRELESS SENSOR NETWORKS BY MERGING RIGID CLUSTERS

Saroja Kanchi

Department of Computer Science, Kettering University, Flint, MI, USA

ABSTRACT

Localization of Wireless Sensor Network (WSN) is the problem of finding the geo-locations of sensors in a sensor network deployed in various applications. Given the prolification of sensors in various applications, the localization and tracking of sensors have received considerable attention. Properties of rigidity and flexibility of the underlying graph of the WSN have been studied as a means of determining the localizability of the nodes in the WSN. In this paper, we present a new 3-merge technique for merging three rigid clusters of a network graph, into larger rigid cluster and we use this algorithm for finding maximal localizable regions within the WSN. We provide simulation results on random deployments of WSN to prove that this technique outperforms previously known algorithms for finding maximal localizable subregions. Moreover, simulation results show that the number of anchors needed to localize the entire WSN decreases due to finding large localizable regions.

KEYWORDS

Wireless Sensor Network, localization, rigidity, cluster, merging

1. INTRODUCTION

Wireless sensor networks are a collection of sensor nodes deployed in various applications including environmental monitoring, search and rescue missions, autonomous driving, target tracking, healthcare monitoring, forest fire detection etc.[13][7][16][17]. Awareness of the exact location of the sensors is cruicial to the success of these applications. Once deployed, in a majority of these applications, the sensors move after deployment and therefore predetermining the location of the sensors is not practical. Moreover, it is not always possible to equip the sensors with GPS due energy consumptions and obstructions in indoor applications. Determining the ge-locations of sensors is the problem of *localization* of a WSN.

There have been several approaches to localization depending the capability of the sensor nodes to obtain various information about the context it is in, and whether the algorithm is centralized or distributed[9]. The *range-based* approaches assume that sensors can find the distance to neighborng sensors within their sensor radius using RSSI signal strength, or time difference of arrival (TOA) between radio signals. In addition, angle of arrival (AOA) of a signal [10], can be used determine the location of the sensors. In range-free approaches, where distance betweeen sensors in not known, [2][12], number of hops is used for localization. Many approaches assume that there are specialized anchors whose location is known, in cases where limited number of GPS equipped sensors are available. In range-free localization, number of hops to an anchor is used as a means of locating sensors.

Natarajan Meghanathan et al. (Eds) : CSEIT, CMLA, NeTCOM, CIoT, SPM, NCS, WiMoNe, Graph-hoc - 2019
pp. 321-330, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91326

In self-localization, nodes localize using distribued computation[9][11], by exchanging information with surrounding nodes. The geometric property of location of nodes dictates that given a set of nodes whose locations are known and an unlocalized nodes whose distance to three localized nodes are known, the location of the unlocalized node can be uniquely determined. The process of thus growing localized set of nodes by spanning out the localized nodes is called *trilateration*. Trilateration [1],[18] is commonly used as a means of localizing nodes, and often a variation of bilateration is used to find location of nodes. Moreover localization is assisted by a mobile anchor or mobile robot that help add missing distances between sensor nodes [15][19].

In centralized approach where each node sends its data to a centralized server, the distance map or any other information provided can be used for localization. The MDS-MAP [6] technique finds the missing distances using shortest path algorithm and uses distance matrix for localization. It turns out that given a distance map between nodes, finding the exact geo-locations of nodes is closely related to the problem of rigidity of the underlying network graph. Therefore finding large rigid subgraphs within a WSN is extremely useful in localizing large number of sensor nodes. Therefore there has been considerable interest in using rigdity for localization. [3][9]. Recently, Erin [4] has proposed a new graph invariant for graph rigdity, namely redudancy index and rigdity index. There exists a unique realization of the graph onto 2D plane if and only if the given the WSN is uniquely localizable. While checking if a graph is globally rigid is polynomially solvable, finding locations of the nodes in a rigid graph is NP-Complete. Using our polynomial time algorithm, large globally rigid subregions can be found in the network each of which are localizable.. Note that actual realization of this lower bound would require 3 anchors per rigid retion, and using MDS-MAP algorithm for each rigid region and merging the local maps to obtain a global map. Our experimental results indicate that this new 3-merge technique localizes large number of nodes in any randomly deployed WSN.

It is shown that even in sparse networks, a large percentage of nodes can be localized with as few as 3 anchor nodes. The paper is organized as follows. In Section 2, we provide the details of rigidity theory of graphs. In Section 3, we provide the new 3-merge technique used for finding This technique extends the 2-merge technique given in [8]. In Section 4, we present the localization algorithm using the new theorem. In Section 5, we present the results of simulation.

2. GRAPH RIGIDITY AND LOCALIZATION

In this section, we introduce the theory in network localizability and rigidity. A detailed description can be found in [3].

Given a network graph WSN sensor nodes 1..n and distances between a subset of node pairs, the network localization problem is to determine the unique locations of the nodes such that the eucledean distance between the localtion of sensor node i and sensor node j is the distance between sensor nodes i and j of the sensor network.

We model the network as a graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ denote the sensor nodes of the network and an edge $(v_i,v_j) \in E$ exists if the distance between v_i and v_j is known. The edge weight w_{ij}, denotes the distance between the nodes v_i and v_j. The network localization problem is to determine the locations of V such that the eucledean distance between the vertex locations is equal to the edge weight w_{ij}, for each edge $(v_i,v_j) \in E$. If under the given constraints, there is only one position for each node, then the network is localizable. The problem of localization is to find the unique location of each vertex subject to given distance information between vertices.

The network localization problem is closely related to the Euclidean graph realization problem. A *framework* of a graph G is a mapping of vertices of G onto 2D plane, such that distance between two vertex placements precisely equal the edge weight of the corresponding edge in G. We can think of this framework as bar and joint framework, where bar corresponds to edges and joint corresponds to vertices. The bar-and-joint framework is *generically rigid* if it has only trivial deformations, as shown in Figure 1, e.g., translations and rotations. Laman characterized rigidity combinatorially [10].

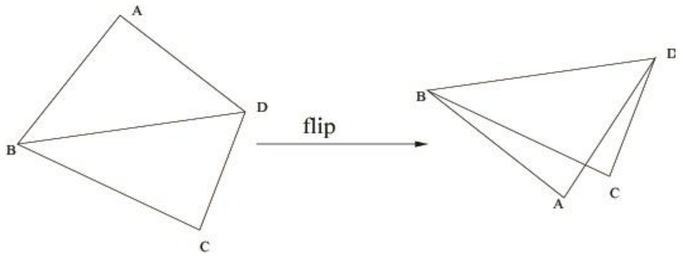

Figure 1. Generically Rigid Graph

Laman's theorem can be intuitively explained as follows. For a two dimensional graph with n vertices, the positions of its vertices have 2n degrees of freedom, of which three are the rigid body motions. Therefore graph is rigid if there are $2n - 3$ constraints. If each edge adds an independent constraint, then $2n - 3$ edges should be required to eliminate all nonrigid motions of the graph. Clearly, if any induced subgraph with n vertices has more than $2n - 3$ edges then these edges cannot be independent which leads the following version of Laman's theorem [10].
Theorem 1. The edges of a graph $G = (V, E)$ are independent in two dimensions if and only if no subgraph $G' = (V', E')$ has more than $2n' - 3$ edges, where n' is the number of nodes in G'.

Corollary 1. A graph with $2n-3$ edges is generically rigid in two dimensions if and only if no subgraph G' of G has more than $2n' - 3$ edges, where n' is the number of nodes in G'.

A framework (G , p) is *globally rigid* if, the distance between every pair of nodes is preserved for different framework realizations, and not just those defined by the edge set. If a graph $G = (V, E)$ is generically rigid but contains more than 2n-3 edges, then G is called a *redundantly rigid* graph. For such a graph, $G - e$ is rigid for all $e \in E$. An edge is called a *redundant edge* if graph remains rigid after its removal. It is known that G has a unique generic realization, i.e globally rigid in 2-space if and only if G is 3-connected and redundantly rigid [22] . Therefore, in order to find unique locations of nodes in a network, we need the underlying graph to be globally rigid and vice-versa.

The problem determining localizability thus reduces to the problem of finding global rigidity. The globally rigid subregions of a graph become localizable and vice versa.

3. ALGORITHM FOR FINDING RIGID CLUSTERS

In this paper, we set out to find maximal localizable subregion within a network by finding the maximal subregions that are globally rigid. This is done by first finding small globally rigid regions within a network and annexing nearby globally rigid regions.

The algorithm we use for checking redundant rigidity is the polynomial time pebble game algorithm proposed by Jacobs[5]. The graph's 3-connectivity property is easily checked in polynomial time.

Given a graph G = (V,E), we define a R_i as subregion of G if R = (V_R, E_R)is globally rigid. In the theorems below, we provide techniques that merge two or three globally regid regions into larger globally rigd regions. The following theorem [8] provides a technique called 2-merge, for merging two globally rigid regions.

Theorem 2: Given globally rigid graphs $R_1 = (V_1, E_1)$ and $R_2 = (V_2, E_2)$, the graph formed by merging the two regions, $R_{2\text{-merge}} = (V_1 \cup V_2 , E_1 \cup E_2 \cup E')$ consisting of additional edges E' described in one conditions (a) to (d) is a globally rigid graph.

a. There are three or more vertices in common between V_1 and V_2. The additional edges consist of edges with one end point in V_1 and other in V_2, and E' could be empty.
b. There are two vertices in common between V_1 and V_2, and there is at least one additional vertex in V_1 that has at least one edge connecting to a vertex V_2
c. There is one vertex in common between V_1 and V_2, and there are at least two other vertices that each have at least one edge connecting to a different vertex in V_2
d. There are no vertices in common between V_1 and V_2, and there are at least 3 vertices in V_i (i=1,2) each have an edge connecting to a different vertex in V_j ($j \neq i$) and there are at least 4 edges between vertices of V_1 and vertices of V_2.

The proof can be found in [8].

In this paper, we provide a technique, 3-merge, for merging three globally rigid regions into a single globally rigid region.

Theorem 3: Given globally rigid graphs $R_1 = (V_1, E_1)$, $R_2 = (V_2, E_2)$ and $R_3 = (V_3, E_3)$ the graph formed by merging the three regions, $R_{3\text{-merge}} = (V_1 \cup V_2 \cup V_3, E_1 \cup E_2 \cup E_3 \cup E')$ is globally rigid if there are 7 edges connecting the three graphs in such a way that no two regions have more than 4 edges between them and there are at least 3 vertices in each region that have an edge connecting to another region.

Proof: We will prove that the graph $R_{merge} = R_1 \cup R_2 \cup R_3 \cup \{e_i, i = 1, 7\}$ is a globally rigid graph. Note that each R_i i =1,3 are 3-connected and redundantly rigid by Corollary 1. We will prove that the graph $R_{3\text{-merge}}$ is also 3-connected and redundantly rigid.

Since each R_i is 3-connected, there are three vertex disjoint paths between any two vertices with the same R_i, i =1,3. Therefore, WLOG, it is sufficient to prove that there are three vertex disjoint paths from one vertex v_i of R_1 to v_j of R_2. Since there are 7 edges between the three regions, if there are no edges between R_1 and R_2, there must be at least 4 edges between R_1 to R_3 or R_2 to R_3 and this is not the case. Therefore, there is at one edge between R_1 and R_2.

Also, note that there are three vertices in R_1 which have edges with the other endpoint in R_2 or R_3. If three of these edges are between R_1 and R_2, then we have three vertex disjoint paths between any vertex of R_1 and any vertex of R_2. If there are two vertex disjoint paths using direct edges, then there must be a thrid vertex in R_1, connects to R_3. Since there are three edges from R_3 to R_2, (since otherwise the total number of edges between three regions will be less than 7) ,we can use one of the paths from R_2 to R_3, to find the third path from a vertex in R_1 and a vertex in R_2. Thus proving 3-connectivity of $R_{3\text{-merge}}$

To prove redundant rigidity, we will show that removal of any edge leaves the graph generically rigid. Clearly, each graph R_1, R_2, and R_3 is redundantly rigid, which means that removing any edge from any of the graphs, the remaining graph contains $2n_1$-3, $2n_2$-3 and $2n_3$-3 edges spanning the n_1, n_2 and n_3 vertices such that each of these are independent edges in R_1, R_2 and R_3 respectively. We will prove that removing an edge from $R_{3\text{-merge}}$, still leaves an independent set edges of size $2(n_1+n_2+n_3)$ -3 edges. Removing any one of the 7 cross edges, the remaining graph contains $2(n_1+n_2+n_3)$ -9 + 6 = $2(n_1+n_2+n_3)$-3 edges. We will prove that the graph containing the independant edges from each region and the six cross edges, forms an independant set of edges for the merged region.

Note that by Theorem 2, a graph G with n vertices and 2n-3 edges is an independent graph (i.e all of its edges are independent) if there is no subgraph of G, of k vertices with more than 2k-3 edges. Let us consider subgraph S of $R_{3\text{-merge}}$-{e} where e is any cross edge. If the S contains no cross edges, then S is independent due Corollary 1. Consider a subgraph that contains all of the six cross edges. Any subgraph that includes all of these 6 edges, there no more than 2_{k_1}-3, 2_{k_2}-3 and 2_{k_3}-3 in each of the subraphs. Therefore, there are no more than $2(k_1+k_2+k_3)$-9+6 = 2k-3 edges in the subgraph that includes all of the cross edges, proving redundant rigidity. For a subgraph that includes less than the maximum number of cross edges, the same argument holds

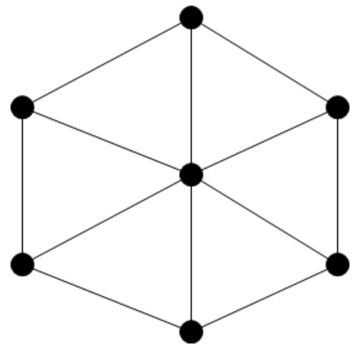

Figure 2. A Smaple Wheel Graph

4. LOCALIZATION ALGORITHM USING RIGID CLUSTERS

Given a WSN graph we perform a centralized algorithm as follows:

1. Find the one-hop globally rigid regions by considering the one-hop neigbors of each vertex. These have wheel structures and they might have vertices in common. See Figure 2.

Repeat steps i) to iii) until no additional merges are possible.

i) To merge two one-hop rigid regions, we use 2-merge of Theorem 2 with three common vertices we find one the conditions of (a). This step is repeated until no additional 2-merges are possible,

ii) The resulting rigid regions from Step i) are merged using 2-merge with edges in common by looking to see if conditions (b), (c) and (d) of Theorem 2 hold. Again this step is repeated until no additonal merges are possible.

iii) The resulting rigid regions from Step ii) are merged using 3-merge algorithm of Theorem 3, looking for three regions for which conditions of 3-merge hold. This step is repeated until no additional merges are possible.

2. Once the large rigid regions are thus formed, we use 3 anchors in each rigid region to localize the rigid regions.

5. SIMULATION RESULTS

Simulation was performed on Matlab, using 100 nodes on a 100 by 100 square foot area with various radii from 12 to 22. The nodes were uniformly distributed over the area. The results in

Figure 3, show that even for really sparse networks with radius as low as 17, significant number of nodes belong to rigid regions and can be localized using three anchors per region.

Figure 3 shows that using 3-merge the number of anchors needed for localizing all localizable nodes dramatically decreases when the radius goes from 12 to 22 and 3 anchors suffice for localizing more than 98% of the nodes when the radius is 22, as indicated in Figure 4. Figure 5 demonstrates that this technique finds really large rigid subgraph and largest rigid subgraph size constains most of the nodes for networks of radious 22. Figures 6 and 7 demonstrate that even for a sparse graph, the number of rigid regions that the algorithm finds are numerous as outlined by the cyan edges. Figures 8 and 9 demonstrate that for dense graph with a radius of 22, all rigid regions are merged into single rigid region making the entire network localizable with just 3 anchors.

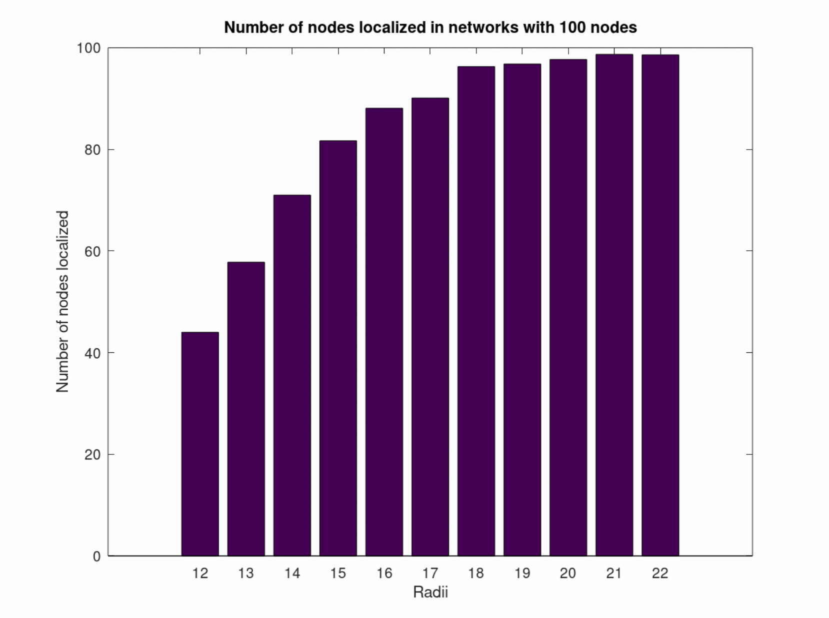

Figure 3: Number of nodes in rigid regions with 3-merge

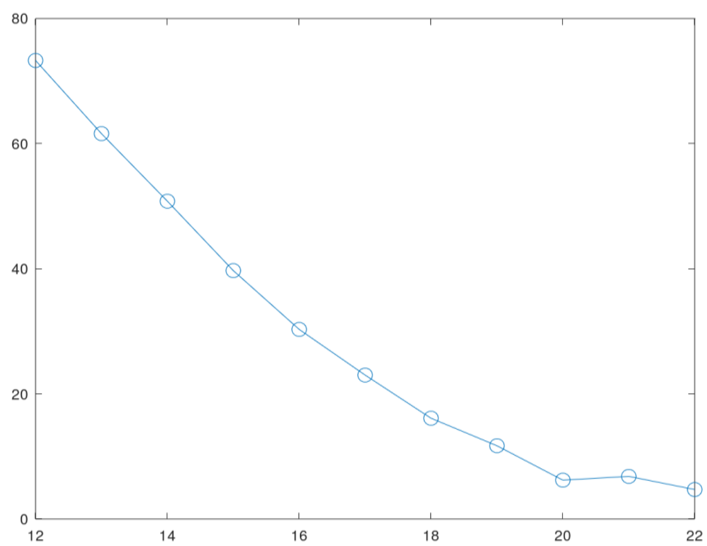

Figure 4: Number of anchors needed to localize the nodes in rigid regions

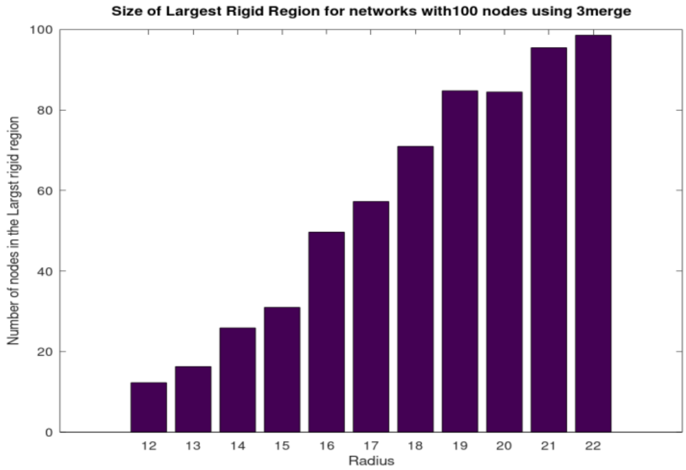

Figure 5: Size of largest rigid region when network regions merged using 3-merge

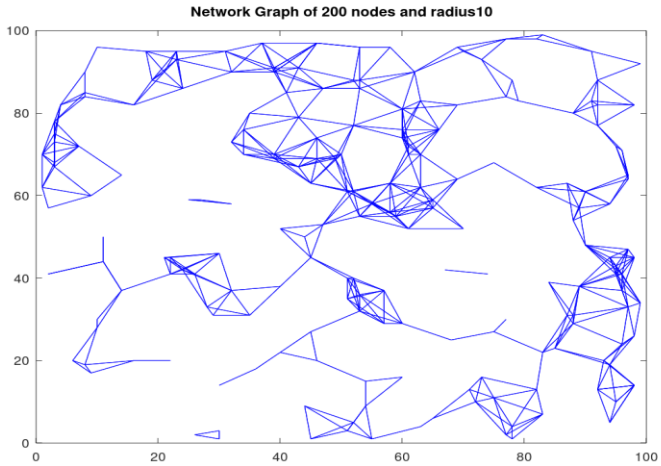

Figure 6: A sparse network graph with radius of 10

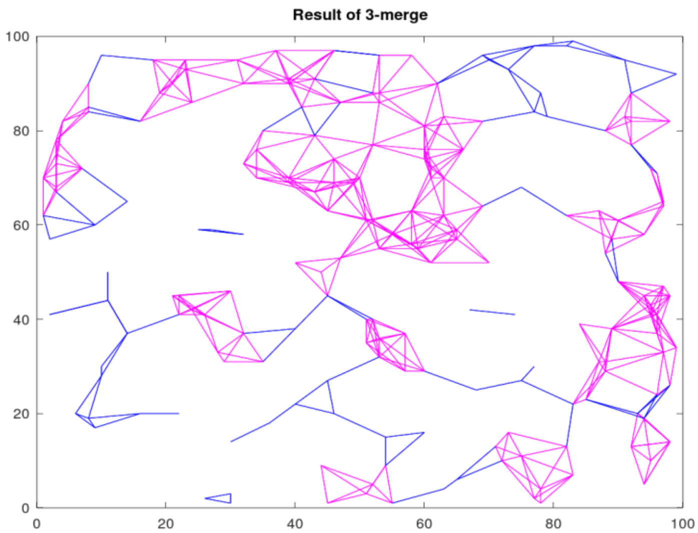

Figure 7: Rigid regions found in the graph in Figure 4 using 3-merge algorithm

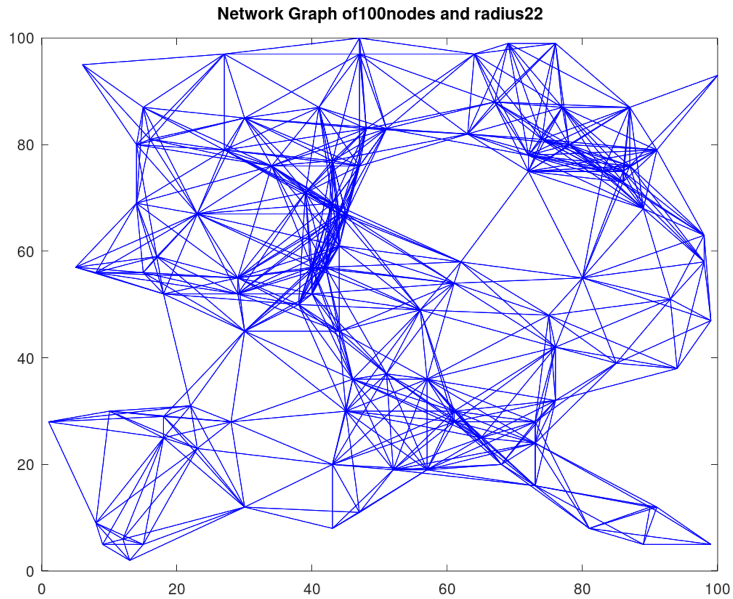

Figure 8: Network with 100 nodes and radius of 22

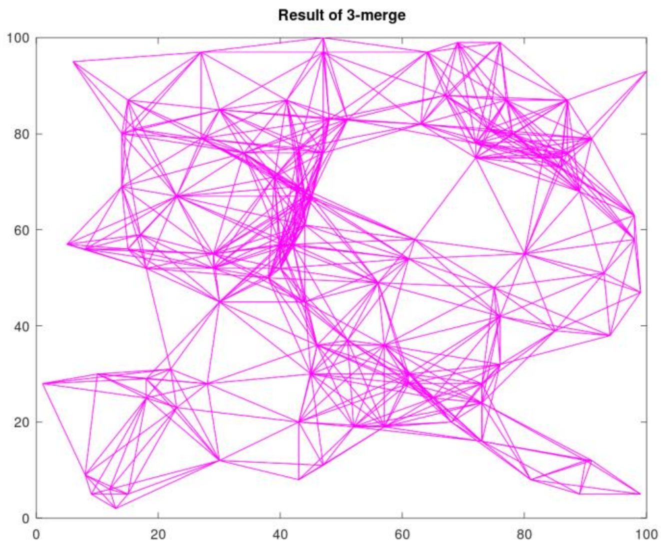

Figure 9: Rigid regions found in the graph in Figure 6 using 3-merge algorithm

6. CONCLUSION

The paper presents a new theorem for merging two rigid regions into a single rigid region for a graph. This theorem is used to find large rigid regions in a network graph, starting with wheel graphs and merging them using 2 merge algorithm first and then merging three regions at a time that obey the conditions of the theorem. The simulation results show that when the radius of the network graph is 19 feet or above in a 100 by 100 feet^2 network the number of nodes localized is over 80%. When the network is sparse, it is important to note that the property is less likely to be found between three regions due the 7 edge requirement between regions. It would be interesting to find out what is the number of rigid clusters that can be merged in a sparse network graph using merging algorithms, after which no significant increase can be found in size of the largest rigid region.

ACKNOWLEDGEMENTS

The author acknowledges Charles Welch for discussions on this topic.

REFERENCES

[1] Chi-ChangChen, Chi-YuChang & Yan-NongL (2013) "Range-Free Localization Scheme in Wireless Sensor Networks Based on Bilateration", International Journal of Distributed Sensor Networks , Article ID 620248, 10 pages

[2] Deepak Prashar & Kiran Jyoti, (2019) "Distance Error Correction Based Hop Localization Algorithm for Wireless Sensor Network" Wireless Personal Communications 106, pp.1465–1488

[3] T. Eren, O. Goldberg, W. Whiteley, Y. R. Yang, A. B. Morse, B. Anderson & P. Belhumeur, (2004) "Rigidity, computation, and randomization in network localization, in: Proceedings", IEEE INFOCOM Vol. 4,pp. 2673–2684.

[4] T. Eren (2016), "Graph invariants for unique localizability in cooperative localization of wireless sensor networks: Rigidity index and redundancy index" Ad Hoc Networks, 44, pp 32-45

[5] D. J. Jacobs & B. Hendrickson (1997) An algorithm for two-dimensional rigidity percolation: the pebble game, Journal of Computation Physics 137 (1997) 346–365.

[6] X. Ji, H. Zha, (2004), "Sensor positioning in wireless ad-hoc sensor networks using multidimentional scaling", I, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, Vol 4, pp.2652 - 2661

[7] Jyoti Kashniyal ,Shekhar Verma & Krishna Pratap Singh, (2019) "A new patch and stitch algorithm for localization in wireless sensor networks", ,Wireless Networks" Vol. 25, No 6, pp 3251–3264

[8] Saroja Kanchi & Charles Welch, (2013) ", An Efficient Algorithm for Finding Large Localizable Regions in Wireless Sensor Networks" Procedia Computer Science Volume 19, pp. 1081-1087

[9] Saroja Kanchi & C. Wu, "Distributed Algorithm for Maximal Rigid Region in Sparse Wireless Sensor Networks", (2010) IEEE Asia-Pacific Services Computing, pp.400-404

[10] G. Laman, "On graphs and rigidity of plane skeletal structures", (1970), Journal of Engineering Mathematics, vol 4, pp. 331–340.

[11] Mort Naraghi-Pour & Gustavo Chacon Rojas, (2014) "A Novel Algorithm for Distributed Localization in Wireless Sensor Networks", ACM Transactions on Sensor Networks. Vol 11, pp.1-25.

[12] Dragos Niculescu & Badri Nath, (2003) "Ad Hoc Positioning System (APS) Using AOA", Proceedings - IEEE INFOCOM, pp.1734 – 1743.

[13] Slavisa, T, (2017). "Distributed algorithm for target positioning in wireless sensor networks using RSS and AoA measurements☹. Pervasive and Mobile Computing, 37, pp. 63-77.

[14] Suman Pandey & Shirshu Varma (2016) "A Range Based Localization System in Multihop Wireless Sensor Networks: A Distributed Cooperative Approach" Wireless Personal Communications ,86, pp. 615–634

[15] Soheil Salari, Il-Min Kim, & Francois Chan, (2018) ‘Distributed Cooperative Localization for Mobile Wireless Sensor Networks”, IEEE Wireless Communications Letters, vol. 7, no. 1,

[16] Tan Wang , Hong Ding, Hui Xiong & Linhua Zheng, (2019), “A Compensated Multi-Anchors TOF-Based Localization Algorithm for Asynchronous Wireless Sensor Networks”, IEEE Access, vol 7, pp.64162-64176

[17] Tashnim, J. S. (2016). “Advances on positioning techniques for wireless sensor networks: A survey: “Computer Networks, 110 pp. 284-305.

[18] Jing Wang, R. K. Ghosh & Sajal K. Das, (2010) “Rigidity, Computation, and Randomization in Network Localization” , J Control Theory Appl , 8 (1) , pp 2–1.

[19] Wu, Zhang, Cheng, S. Kanchi (2010), “Rigidity guided localisation for mobile robotic sensor networks”, Journal of Ad Hoc and Ubiquitous Computing, vol 6, 2, pp.114-127.

AUTHORS

Dr. Kanchi is a Professor and the Interim Department Head of Computer Science at Kettering University. She works in the areas of Topological Graph Theory, Wireless Sensor Networks and Distributed Algorithms. She has published several articles in these areas.