International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.7 (2015), pp.329-338
http://dx.doi.org/10.14257/ijmue.2015.10.7.34

Node Deployment Algorithm Based on Improved Steiner Tree

Xuemei Sun*

School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin, China

Bo Yan

School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin, China

**Address correspondence to this author at No. 399 Binshui Road, Xiqing District, Tianjin, China. Postcard: 300387; Tel:+8618502285875; E-mail: tjpu_sun@163.com*

Abstract

Wireless sensor networks are being applied to a wider and wider range and better quality of network service is becoming increasingly important. Node deployment has been a key point of research on development of wireless sensor network. In essence, the problem node deployment of wireless sensor networks calculates the position where sensors are placed in regions to satisfy specific demands of network. This problem has been proved to be a NP-complete one. This thesis proposes node deployment algorithm about wireless sensor networks based on Steiner tree algorithm. On the basis of the triangle Steiner tree algorithm, it considers the two aspects algorithm complexity and practical topology to design improvement strategies. The algorithm can ensure high arithmetic speed and better results to solve node deployment based on network connectivity.

Keywords: *wireless sensor network. node deployment. Steiner tree*

1. Introduction

With expansion of our knowledge, information acquisition, storage, processing and transmission have become an indispensable part of human life. Wireless sensor networks (WSNs) emerge at the right moment by integrating cutting-edge technologies like sensor technology (ST), micro electro mechanical system (MEMS), wireless communication technique (WCT) and distributed information processing technology. Wide development and application of WSNs arouse a number of researchers' interest. As a primary link of WSN design, node deployment has become one of the most fundamental and the most critical problems of research.

Network connectivity is a factor that should be considered first in node deployment [2]. A typical solution to the problem is minimum spanning tree (MST). For instance, Hisham M. Almasaeid [3] adopted thoughts about MST algorithm to design node deployment algorithm about WSNs in his article. Sookyoung Lee [4-6] utilized Sookyoung Lee [4-6] utilized minimum tree to solve node deployment algorithm based on Mesh model. However, obvious problems exist in selection about points representing independent network in the algorithm. Besides, true form and connection mode of independent network are not considered. Additionally, deployment method of Mesh network model reduces accuracy of node deployment largely. Fatih Senel [7] also designed node deployment algorithm about WSNs by the triangle Steiner tree algorithm a variant of Steiner tree algorithm. Compared with the algorithm proposed by Sookyoung Lee, the algorithm has better practical application effect and is based on irregular network models.

ISSN: 1975-0080 IJMUE

Nevertheless, the algorithm itself has high time complexity, which makes it take much time. In addition, its deployment effect is good only when communication radius is smaller. Under the situation that communication radius is larger, its deployment effect is not ideal.

On the basis of these, this thesis specifically studies Steiner tree algorithm and proposes a node deployment algorithm based on improved Steiner tree algorithm. The method can reduce cost of network deployment and decrease running time of the algorithm largely on the premise that network connectivity is ensured.

2. Overview on the Triangle Steiner Tree Algorithm

Steiner tree may be traced back to Fermat problem, *i.e*., with respect to any three points A, B and C in a plane, find a point P to make sum of its distance to the three points A, B and C minimum. The point P is called Fermat point. The problem Steiner tree may be descried as follows, *i.e*., considering n points in a plane, find a point set E to connect these n points. The solution to the problem is called Steiner minimum tree and points in E are called Steiner points.

2.1 The Triangle Steiner Tree Algorithm and Related Definitions

Definition 1: weight of side (u,v) is set as the number of relay nodes needed to connect points u and v. The computational formula is shown in (1).

$$W_e(u,v) = \left\lceil \frac{|uv|}{R} \right\rceil - 1 \tag{1}$$

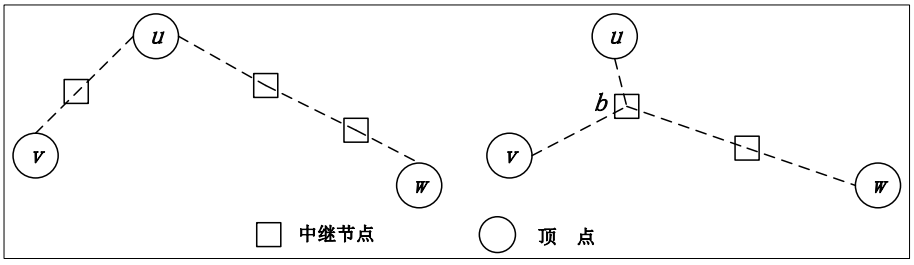
Where R represents communication radius of relay nodes and |uv| stands for Euclidean distance between u and v.

Definition 2: regarding a triangle constituted by any three points u, v and w in the Figure, there are two easy to make the triangle Ti become a connected graph. (1) Use MST to connect the three points u, v and w, as shown in Figure. 1(a). Then, weight of the triangle is defined as sum of two sides' weight. The computational formula is shown in (2):

$$W_{mst}(T_i) = \left\lceil \frac{|uw|}{R} \right\rceil + \left\lceil \frac{|uv|}{R} \right\rceil - 2 \tag{2}$$

(2) Find Fermat point b of the three points (u,v,w) and connect sides (u,b), (v,b) and (w,b), as shown in Figure. 1. Then, weight of the triangle is defined as sum of the three sides' weight. The computational formula is shown in (3):

$$W_c(T_i) = \left\lceil \frac{|bw|}{R} \right\rceil + \left\lceil \frac{|bv|}{R} \right\rceil + \left\lceil \frac{|bu|}{R} \right\rceil - 2 \tag{3}$$



(a) (b)

Figure 1. Connecting Methods about the Triangle Ti

Definition 3: weight of the triangle Ti = (u,v,w) is expressed as Formula (4):

$$W_t(T_i) = \min\{W_{mst}(T_i), W_c(T_i)\} \tag{4}$$

Definition 4: covered point: as Point v is connected to a connected component, the point is not isolated any more. Then Point v is called a covered point. Uncovered point: Point v is not connected to any other node, so it is called an uncovered point.

Definition 5: connected component: as for some apexes, they may be connected with one another to constitute a connected component by adding additional nodes. Then, apexes in the connected component are called covered points. As shown in Figure. 2, there are two connected components.

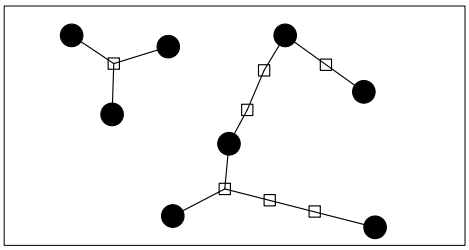


Figure 2. Graphical Representation about Connected Components

According to the foregoing definitions, realization process of the triangle Steiner tree algorithm may be divided into three parts overall, including (1) establishing connected components; (2) uniting connected components in the first part and (3) optimizing MST side in the triangle. See details about the algorithm in Literature [7].

2.2 Application and Limitation of the Triangle Steiner Tree Algorithm

The triangle Steiner tree algorithm itself is effective, but its time complexity is O (n4). In accordance with time complexity of the algorithm and experiments, it is found that running time of the algorithm increases with the number of apexes and the more the communication radius is, the more effective the triangle Steiner tree algorithm will be. However, as communication radius increases, its results do not have advantages. The triangle Steiner tree algorithm put forward by Fatih Senel is mainly applied to independent parts. These independents parts are treated as terminals in theses, which is an unreasonable practice. The reason for this is that ways to select terminals may affect results of the algorithm and size and shape of independent parts have significant influence on terminals.

Aiming at the foregoing defects of the triangle Steiner tree algorithm, this thesis will improve it in three aspects: (1) improving the algorithm to reduce its running time; (2) proposing recovery algorithm to make overall computational results of the algorithm approach practical environment to a larger extent and (3) improving the algorithm to make it have good results when communication radius of sensor nodes increases.

3. Description about the Node Deployment Algorithm Based on Improved Steiner Tree

Directing at features about node deployment of WSNs, this paper puts forward new improvement strategies based on the triangle Steiner tree algorithm, *i.e.*, 'gathering parts into a whole', selection of representative points and recovery strategy, and Node Deployment based on Improved Steiner Tree Algorithm (DISTA). Figure. 3 shows an overall flow chart about the algorithm.

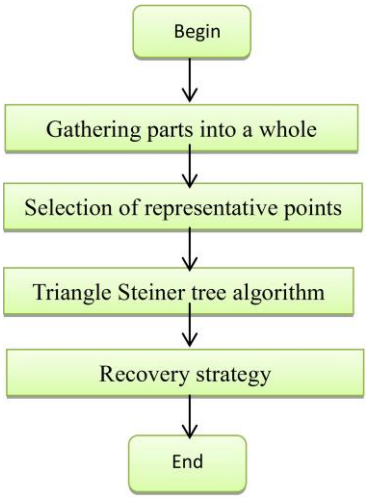


Figure 3. Overall Framework of the Improved Steiner Tree Algorithm

3.1 DISTA Detailed Design

This section will describe specific detailed designs about 'gathering parts into a whole', selection of representative points and recovery strategy in DISTA in detail.

1. 'Gathering parts into a whole'

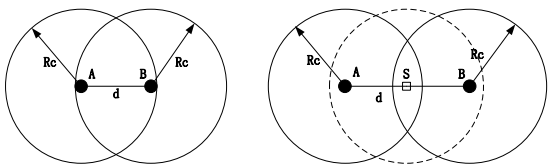
'Gathering parts into a whole' judges positional relations among sensor nodes by considering their position and communication radius (expressed as Rc) and divides nodes in the Figure into one or several partitions (PAs). The following three cases are mainly considered:

Case 1: sensor nodes may be connected with one another directly, *i.e.*, Euclidean distance between two sensor nodes is less than or equal to Rc of sensor nodes, as shown in Figure. 4(a). In this case, sensor nodes that may be connected with one another are connected to constitute a small partition.

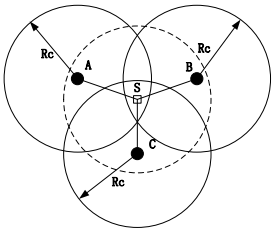
Case 2: it is possible to ensure two sensor nodes can communicate with each other by adding a node between them, as shown in Figure. 4(b).

Case 3: we may ensure three sensor nodes can communicate with one another by adding a node among them, as shown in Figure. 4(c). Actually, this case may appear in the triangle Steiner tree algorithm. By looking for this case, we hope to reduce the number of nodes in the triangle Steiner tree algorithm.

It is supposed that there are 10 nodes in the original drawing. If the triangle Steiner tree algorithm operates, we need calculate and compared C_{10}^{3} triangles. By the strategy 'gathering parts into a whole' and selection of specific representative points, we make reduce the number of nodes in the triangle Steiner tree algorithm to 5 so that we only need calculate and compare C_{5}^{3} triangles. In doing so, running time of the algorithm is reduced largely.



(a) (b)



(c)

Figure 4. A Schematic Diagram about the Strategy 'Gathering Parts Into a Whole'

2. Selection of representative points

With respect to PAs obtained by 'gathering parts into a whole', we use specific strategies to choose their representative points. Description is shown as follows. Choose two points A(x1,y1) and B(x2,y2) in a small partition. |x1-x2| is maximum in x difference value of any two points in the small partition. Use mid-value of the two points' x value as x value of the representative point P(x,y), *i.e.*, x=(x1+ x2)/2. Similarly, choose two points C(x3, y3) and D(x4, y4) with maximum y difference value in the small partition. y value of the representative point P is defined as (y3+ y4)/2. Point P obtained by calculation is a representative point of the partition.

It is supposed that a partition PA is obtained (as shown in the following) by 'gathering parts into a whole'. The partition is a connected component composed of 10 nodes and 3 additionally relay nodes. Then, x and y values of the representative point P in the partition is (x1+x10)/2 and (y2+ y6)/2, respectively. In the example Figure. 5, the square and red point P is a representative point of the partition.

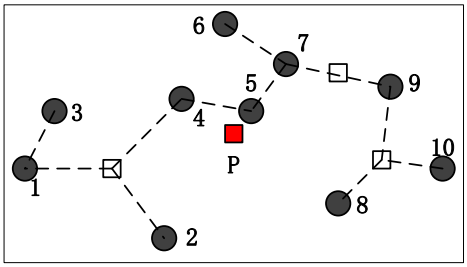


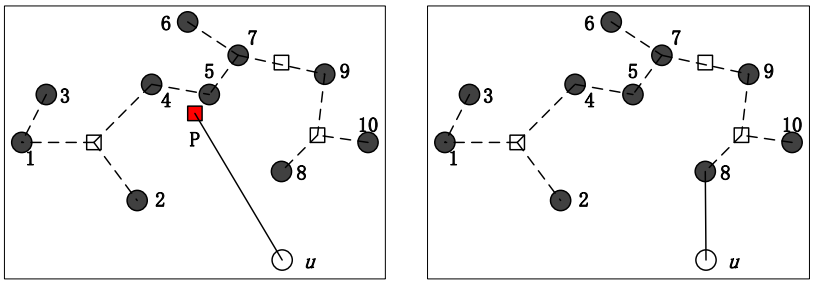
Figure 5. An Example about Selection of Representative Points in the Partition PA

3. Recovery strategy

Via the foregoing two strategies, *i.e.*, 'gathering parts into a whole' and selection of representative points, the number of nodes taking part in the triangle Steiner tree algorithm in the graph G(V,E) may be reduced largely. However, existence of virtual points made deployment results and node connection methods divorced from reality. As shown in Figure. 5, the representative point P is a virtual point and does not exist in reality. Recovery strategy aims at realizing node connectivity in practical environment, *i.e.*, removing virtual representative points and using practical nodes to realize connectivity of a whole network.

Recovery strategy is described as follows specifically. The representative point P of the partition PA and a node u (u∈ V) outside the PA are connected. Then, look for v (v∈ PA) the node that is the closest to the node v in the PA, delete the side (P,u) and connect with the side (u,v). As shown in Figure. 6, the representative point P and u a point outside the

PA are connected with each other by deploying additional nodes and recovery strategy deletes additional nodes on the side (P,u) and connects the side (8,u).



(a) (b)

Figure 6. A Schematic Diagram about Recovery Strategy

3.2 DISTA Algorithm Flow

According to foregoing statement, realization process of DISTA may be described as follows:

Input: WSN topology G, Rc communication radius of sensors and Area size of the region where networks locate;

Output: node deployment satisfying connectivity conditions, *i.e.*, a Steiner tree;

Begin

Divide the graph F into m Pas: 'gathering parts into a whole';

Delete each node belonging to the partition PA from the set V. Then, we may get V'=V-{v|v∈ PA};

For i=1 to m do

Choose the representative point p->P of the partition from the ith partition PA. //P represents a set of each partition's representative points V' ∪ P->V';

Operate Algorithm 1 for nodes in V', *i.e.*, carry out the triangle Steiner tree algorithm;

Implement recovery strategy for the obtained Steiner tree;

Output the Steiner tree obtained by using recovery strategy;

End

4. Simulation Results and Experimental Analysis

In order to verify feasibility and effectiveness of DISTA, this section will evaluate performance of the algorithm by simulation experiments. Meanwhile, other two representative algorithms are compared here to explain effectiveness of the improved algorithm. According to MST and basic triangle Steiner tree [21], corresponding node deployment algorithm about WSNs is designed, which is written as DMST and DTST, respectively.

As Literature [21] does not provide related data for comparison, a way to generate network node position randomly is used. According to setting about WSN models above, a 500m*500m area is assumed here, where n randomly generated sensor nodes are distributed. Each sensor node has a unique identification number. Two groups of experiments are set in this thesis. Each group of experiments does not consider perception radius of sensor nodes. Parameter setting of the first group of experiments Set I is shown as follows: set Rc=10, value n by the increment 10 in the range [10,100] and analyze changes in the number of additionally deployed nodes as the number of original nodes varies. As for the second group of experiments Set II, value Rc communication radius of sensor nodes by the increment 5 in the range [5,70] when n=50. In this case, the relationship between the number of additionally deployed nodes and communication radius of sensor nodes is analyzed.

4.1 Analysis about Node Deployment Results of DISTA under the Condition of Set I

In the 500m*500m area, generate 10, 20, and 100 sensor nodes randomly according to Set 1. In another word, generate 10 kinds of original sensor networks accordingly. Besides, it is supposed that communication radius of sensor nodes is 10m and communication radius of sensor nodes that are added by DISTA is also 10m. Then, implement 30 experiments on each kind of sensor networks. The number of sensor nodes that need be deployed additionally is shown in Table 1-1.

With respect to node deployment algorithm based on MST (DMST), Table 1 shows node deployment algorithm based on triangle Steiner tree (DMST) and node deployment algorithm based on improved Steiner tree (DISTA), deployment results of DISTA and DTST are superior to DMST when the number of original nodes is small under the condition of Set I. However, as the number of original nodes increases, DTST is obviously inferior to DMST. Compared with the other two algorithms, DISTA has superiority when n the number of nodes is large, *i.e*., the number of sensor nodes that need be deployed additionally is smaller than the other two algorithms. When n the number of original nodes is small, it degrades into DIST, *i.e*., the number of sensor nodes that need be deployed additionally is approximate to that of DTST. Overall, ISTA is of more usability than the other two algorithms.

Table 1-1 Comparison about Deployment Results among Three Different Algorithms under the Condition of Set I

n	DMST	DTST	DISTA
10	101.34	95.6	95.75
20	145.55	140.65	141.3
30	174.22	172.1	172.75
40	197.41	200.2	199.7
50	218.82	222.4	218.4
60	241.94	231	235.15
70	249.98	252.95	249.5
80	264.7	263.8	263.5
90	277.15	277	273.1
100	292.55	290.55	288.15

When n the number of original nodes reduces, the reason why DISTA degrades into DIST is that Rc=10 in the 500m*500m area. When n is small, there is nearly no intersection of communication areas among original sensor nodes, which indicates that all of the improvement strategies including ‘gathering parts into a whole’, selection of representative points and recovery strategies cannot be used. This directly makes DISTA degrade into DIST. As the number of original nodes increases, it means there may be intersection of communication areas among original sensor nodes, which implies improvement strategies can be used. In this way, advantages of DISTA are shown. As a whole, experiments show DISTA has more superiority under the condition that there is intersection of communication areas among original sensor nodes. Then, superiority, effectiveness and usability of DISTA are proved overall.

4.2 Analysis about Node Deployment Results of DISTA Under the Condition of Set II

According to initial setting conditions about network environment in Set II, generate 50 sensor nodes in the 500m*500m region randomly. In other words, they serve as original sensors contained in initial environment of sensor networks. Besides, it is

assumed that communication radius of sensor nodes changes in the range of 5m, 10m, 15m and 70m. Moreover, communication radius of sensor nodes added by DISTA is the same as that of original sensor nodes. Carry out 30 experiments on each kind of sensor networks. The number of sensor nodes that need be deployed additionally is shown in Table 2.

Considering the three deployment algorithms, Table 1-2 shows that deployment results of DISTA are not ideal compared with the other two algorithms but approximate when communication radius Rc is small. Nevertheless, as Rc of sensor nodes increases, DISTA has obvious advantages compared with the other two algorithms, *i.e.*, the number of sensor nodes that need be deployed additionally is smaller than the other two algorithms. Similarly, when Rc of sensor nodes is small, DISTA may degrade into DTST; when Rc is large, DISTA may have more superiority than DTST. However, when Rc is neither too big nor too little, results of DISTA are inferior to that of DTST. Overall, DISTA is of more usability than the other two algorithms when Rc is or large small.

Table 1-2 Analysis about Deployment Results of the Three Algorithms under the Condition of Set II

Rc	DMST	DTST	DISTA
5	445.9	440.5	438.95
10	210.3	208.4	210.1
15	133.2	132.8	133.6
20	92.9	93.4	97.35
25	69.7	69.2	74.4
30	54.7	55.4	59.65
35	42.2	44.1	50.7
40	33.1	34.7	35.8
45	28.2	29.1	30.25
50	22.7	23.7	22.65
55	18.8	20.2	16.9
60	15.2	16.4	13.35
65	11.6	13.3	10.1
70	8.9	10.9	7.75

In the foregoing case, the reason why DISTA degrades into DTST when Rc is small is the same as that in the first group of experiments. Similarly, analysis about the situation that DISTA has superiority when Rc of sensor nodes is large is also the same as that in the first group of experiments. Hence, no further explanation will be given here.

4.3 Analysis about DISTA Effectiveness

DISTA was put forward based on differences between DTST and DMST. Thus, the result obtained by analyzing node deployment results of three algorithms is that DISTA is approximate to DTST under a certain condition. On the contrary, DISTA is approximate to DMST sometimes. This section will compare DISTA and DTST.

First of all, compare running time of DISTA and DTST. Except for the situation that DISTA degrades into DTST, running time of DTST is shorter than that of DTSt under the other conditions. The reason for this is that Literature [21] states the time complexity of DTST is O (n4), *i.e.*, its running time will increase as the number of nodes taking part in the triangle Steiner tree algorithm grows. However, DISTA reduces the number of nodes in the triangle Steiner tree algorithm by ‘gathering parts into a whole’ too some extent, *i.e.*, DISTA reduces running time to a certain extent.

Secondly, DISTA has more actual utility compared with DTST. The reason for this is that operating environment of actual utility is based on abstract n nodes basically. In other

words, each independent part is treated as a terminal. This practice does not accord with practical application to some extent. Besides, computed results do not correspond to practical demands. After the strategy selection of representative points, DISTA adds recovery strategy. On the one hand, this reduces running time of the algorithm; on the other hand, it enables computational results, *i.e*., node deployment, to have more actual utility. This is a condition that all research should satisfy as much as possible.

In short, DISTA improves DTST, which can not only reduce running time of the algorithm but also have more actual utility and reduce cost of node deployment to some extent. Thus, effectiveness of DISTA is shown.

5. Conclusion

Firstly, this thesis introduces basic theory and steps of the triangle Steiner tree algorithm in detail. Then, it focuses on elaborating a WSN node deployment algorithm based on improved Steiner tree algorithm. The algorithm treats the triangle Steiner tree algorithm as a basis and includes it in framework of the whole improved algorithm. By virtue of three strategies, including 'gathering parts into a whole', selection of representative points and recovery strategy, it improves shortages of the triangle Steiner tree algorithm, such as long operation time and bad results, and reduces cost of overall node deployment. Results of simulation experiments show that DISTA has feasibility and effectiveness and can calculate and obtain results of node deployment rapidly.

Acknowledgements

This paper belongs to the project of the "the National Nature Science Foundation of China", No.61173032; "the Tianjin Higher Education Science and Technology Development Fund", No. 20100806; "the Natural Science Foundation of Tianjin, China", No. 13JCYBJC15500

References

[1] Sitanayah L., Brown K. N. and Sreenan C. J., "A fault-tolerant relay placement algorithm for ensuring k vertex-disjoint shortest paths in wirelesssensor networks," AD HOC NETWORKS, vol. 23, **(2014)**, pp.145-162.
[2] Huang H., Zhang J, and Wang R., "Sensor Node Deployment in Wireless Sensor Networks based on Ionic Bond-Directed Particle Swarm Optimization," Applied Mathematics & Information Sciences, vol. 8 no. 2, **(2014)**, pp.597-605.
[3] Hisham M. A. and Ahmed E. K., "On the minimum k-connectivity repair in wireless sensor networks," IEEE International Conference on Communications, **(2009)**, pp. 1-5.
[4] Sookyoung L. and Mohamed Y., "Optimized Relay Placement to Federate Segments in Wireless Sensor Networks," IEEE Journal on Selected Areas in Communications, vol. 28 no. 5, **(2010)**, pp.742-752.
[5] Sookyoung L. and Mohamed Y., "Recovery from multiple simultaneous failures in wireless sensor networks using minimum Steiner tree," Parallel Distribute Computer, vol. 70, **(2010)**, pp. 525-536.
[6] Sookyoung L. and Mohamed Y., "Optimized relay node placement for connecting disjoint wireless sensor networks," Computer Networks, vol. 56, **(2012)**, pp. 2788-2804.
[7] Fatih S. and Mohamed Y., "Relay node placement in structurally damaged wireless sensor networks via triangular steiner tree approximation," Computer Communications, vol. 34, **(2011)**, pp. 1932-1941.
[8] Liu X. and He D., "Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks," Journal of Network and Computer Applications, vol. 39, **(2014)**, pp. 310-318.
[9] Luo R. C. and Chen O., "Mobile Sensor Node Deployment and Asynchronous Power Management for Wireless Sensor Networks," IEEE Transactions on Industrial Electronics, vol. 59 no. 5, **(2012)**, pp. 2377-2385.
[10] Senel F. and Younis M., "Relay node placement in structurally damaged wireless sensor networks via triangular steiner tree approximation," Computer Communications, vol. 34 no.16, **(2011)**, pp. 1932-1941.

