

An Obstacle-resistant Relay Node Placement in Constrained Environment

Abhishek Verma ¹ and Virender Ranga ²

¹National Institute of Technology Kurukshetra

²Affiliation not available

October 30, 2023

Abstract

Relay node placement in wireless sensor networks for constrained environment is a critical task due to various unavoidable constraints. One of the most important constraints is unpredictable obstacles. Handling obstacles during relay node placement is complicated because of complexity involved to estimate the shape and size of obstacles. This paper presents an Obstacle-resistant relay node placement strategy (ORRNP). The proposed solution not only handles the obstacles but also estimates best locations for relay node placement in the network. It also does not involve any additional hardware (mobile robots) to estimate node locations thus can significantly reduce the deployment costs. Simulation results show the effectiveness of our proposed approach.

An Obstacle-resistant Relay Node Placement in Constrained Environment

Abhishek Verma

Department of Computer Engineering
National Institute of Technology
Kurukshetra, India
abhiverma866@gmail.com

Virender Ranga

Department of Computer Engineering
National Institute of Technology
Kurukshetra, India
virender.ranga@nitkkr.ac.in

Abstract—Relay node placement in wireless sensor networks for constrained environment is a critical task due to various unavoidable constraints. One of the most important constraints is unpredictable obstacles. Handling obstacles during relay node placement is complicated because of complexity involved to estimate the shape and size of obstacles. This paper presents an Obstacle-resistant relay node placement strategy (ORRNP). The proposed solution not only handles the obstacles but also estimates best locations for relay node placement in the network. It also does not involve any additional hardware (mobile robots) to estimate node locations thus can significantly reduce the deployment costs. Simulation results show the effectiveness of our proposed approach.

Keywords—obstacles, constrained, effectiveness, complexity, mobile robots

I. INTRODUCTION.

In the recent years, there has been a rise in the applications of wireless sensor networks (WSNs) [1]. In applications such as space exploration, forest fire detection, combat field reconnaissance, and machine health monitoring, to collectively monitor the area and on look after certain activities, some sensor nodes need to be placed in the network. By using these sensors to control unattended in astute surroundings, it might be attainable to avoid the danger to human life and reduce the value of the appliance. These applications use sensor nodes (SNs) which are battery driven and have limited processing and communication capabilities. After deployment, the sensor nodes set up a network with the target of sharing the data and synchronizing the actions performed. To facilitate such collaboration, nodes must be reachable to each other. Long distance transmission for sensor nodes would be expensive and will exhaust them very rapidly (as energy transmission is proportional to the distance). Thus, relay nodes (RNs) are introduced, which aim to transfer the sensed data through RNs – relay nodes' links from SNs to a base station (BS). The problem of placing minimum relay nodes in the environment so that the entire network is interconnected is shown to be an NP-hard problem [2] and is called relay node placement (RNP) problem. This RNP problem considers two different

architectures, one and two tiers WSNs. In one tiered WSNs, both SN (Sensor node) and RN (Relay node) participate in the routing. While in two tiered WSNs, only RNs are used in routing procedure. The shortcoming of a vast network is that the wireless devices may fail, leaving the WSN disconnected. Thus, it is necessary to introduce fault tolerance in the networks for successful functioning. One approach is to introduce redundant disjoint paths, i.e. there are more than one path between every pair of wireless devices. This ensures that if one path is destroyed, other can keep the ends connected. The RNP is performed in two kinds of environment i.e. constrained or unconstrained. In unconstrained RNP, the RN can be deployed anywhere in the network, while in constrained RNP, there are physical constraints [3] on where RN are to be deployed. This is done to consider the physical limitations of a geographical area, for example, a volcano is present, a huge rock or even mountains or water bodies. These practical assumptions are nearer to the real world problem in idealistic situations.

This paper presents a solution for RNP problem in constrained environment (obstacles). We also discuss about the state of the present research and review a mixture of already published techniques in this area stating their features and limitations. In the section II, we discuss various RNP techniques and obstacle handling techniques present in the literature. Section III, deals with assumptions and system model considered in our approach, in section IV ORRNP strategy is presented. Lastly, In Section V simulations results have been shown.

II. RELATED WORK.

As far as RNP problem is concerned Steiner minimum tree (SMT) [13] network topology is considered as the most efficient approach. There are many research papers present in literature for the solution to RNP problem. RNP techniques are classified in Fig 1.

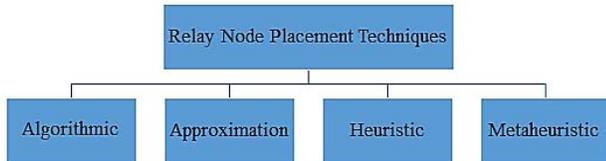


Fig 1. Relay Node Placement techniques

Yu et al. [4] proposed an approach named RNIndoor which targets to provide at least one path among deployed SNs by deploying few additional RNs. This work considers deployment area as floor which contains various obstacles like walls and restricted areas. Approach handles obstacles using structural information of floor and path loss generated from propagation models. The complexity involved in this approach limits it to simple obstacle handling only. In [5] Wang et al. presented a scheme for efficiently deploying SNs. Sensing field of SNs is subdivided into smaller sub-regions based on the shape of field. Further deployment of SNs in these sub-regions is performed. Sensing field is assumed to be an arbitrary shaped region. This work considered obstacles while deploying sensors. The work is limited to a special case which is sensing range is same as transmission range. This problem is limited to a special case and hence cannot be applied to complex obstacles. Verma et al. [6] present a survey for various heuristics and metaheuristics based RNP approaches. Algorithmic and approximation based RNP approaches are surveyed by Sharma et al. [7]. In [8] Ranga et al. proposed an efficient approach to heal partitioned WSNs using RNP. Ranga et al. [9] proposed an optimal nodes selection approach for wireless sensor and actor networks. Prioritized mutual exclusion is used for selecting nodes optimally. Chang et al. [10] presented an obstacle-resistant robot-deployment (ORDD) algorithm. ORDD out performs various other simple robot deployment methods. To govern the movement of robot while the node deployment process some policies are defined which are node placement policy, serpentine movement policy. For handling obstacles, obstacle handling rules and boundary rules are defined. This approach is capable of handling unpredictable obstacles of various complex shapes and sizes. In addition to node placement techniques we review some shape approximation techniques present in literature which we have referred for obstacle estimation and detection. In [11] Urs Ramer proposed an algorithm which iteratively approximated the two-dimensional arbitrary curves to polygons. The ideology behind this work is use small number of vertices lying on plane curve to produce polygons. Fit criterion is chosen as the maximum distance between any curve and approximating polygon. Eddsbrunner et al. [12] presented a computational geometry based scheme that generalizes straight line graphs named “alpha-shapes”. These shapes show much better notion of real complex shapes. Using point set data these shapes are produced. This scheme is capable of approximating any shape in $n \log n$

running time complexity. Obstacle estimation and detection problem is well mapped with this approach.

III. ASSUMPTIONS AND SYSTEM MODEL

A two-dimensional sensing field has been considered for the ORRNP approach. Sensor and relay nodes are assumed to be static in nature after they have been deployed in the constrained environment. Absolute location of SNs is its last known GPS (Global Positioning System) location. A GPS locator is assumed to be equipped with each SN and RN. Sensing range (SR_s) and transmission range (TR_s) of each SN is in ratio ($SR_s:SR_s\sqrt{3}$) respectively. Transmission range (TR_r) of RN is larger than transmission range of SN (TR_s). Sensors whose sensing range overlaps with each other can communicate directly. SNs are assumed to be more in number as compared to RNs, thus a SN to SN connectivity is assumed to work well even if few sensors fail. This assumption minimizes the cost of RNP in WSN. RN can communicate with each node (SN or RN) which falls in its transmission range. In our system we assume sensor to sensor, sensor to relay (vice-versa) and relay to relay connectivity for minimizing the RN count and network deployment cost. Obstacles of arbitrary complex polygonal shapes are assumed to completely block the signal when they fall in transmission path. Proposed approach primarily considers connectivity of nodes.

A. Mathematical formulation

We consider WSNs having SNs and RNs. Transmission range or communication range of SN is assumed to be $r > 0$ and for RN is $R \geq r$. r and R are constant parameters for a particular WSN. This work aims to place least number of RN's in constrained environment (Obstacles) WSNs. We use $\mathcal{D}(p, q)$ to denote the distance between a pair of points (nodes) in the plane, where p, q are two points. Distance can be Euclidean or Rectilinear. Two nodes p, q can interconnect with each other directly if and only if $\mathcal{D}(p, q)$ is less than or equal to the smaller of communication ranges of two nodes. An SN p can communicate directly with other node q (SN or RN) if and only if $\mathcal{D}(p, q) \leq r$. An RN p can interconnect directly with another node q (RN) if and only if $\mathcal{D}(p, q) \leq R$. The above rules result into a communication graph. This graph represents all possible pair of connections between pair of nodes. Fig 2. represents communication graph.

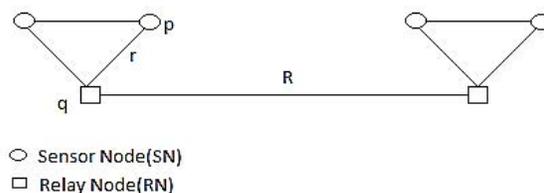


Fig 2. Communication graph

Definition 1: Let P be a set of SNs, Q be a set of RNs, and $R \geq r > 0$ be the corresponding communication ranges of RNs and SNs. The *communication graph* $\text{ComG}(r, R, P, Q)$ resulted by the 4-tuple (r, R, P, Q) is an undirected graph with vertex set $V = P \cup Q$ and edge set E defined as follows. For any two RNs $q_i, q_j \in Q$, E contains the undirected edge $(q_i, q_j) = (q_j, q_i)$ if and only if $\mathcal{D}(q_i, q_j) \leq R$. For an SN $p \in P$ and a node $t \in P \cup Q$ which is either an SN or an RN, E contains the undirected edge $(p, t) = (t, p)$ if and only if $\mathcal{D}(p, t) \leq r$.

Given a set of SNs and a set of candidate locations where RNs can be placed, our aim is to deploy the least number of RNs so that the resulted communication graph by the SNs and RNs is connected.

Definition 2: Let $R \geq r > 0$ be the communication ranges for RNs and SNs respectively. Let P be a set of SNs, and CL be a set of candidate locations where RNs can be placed. A set of RNs $Q \subseteq CL$ is said to be a *possible connected relay node placement* (denoted by P-RNPc) for (r, R, P, CL) if the graph $\text{ComG}(r, R, P, Q)$ is connected. The *size* of the resultant P-RNPc is $|Q|$. An P-RNPc is said to be a *minimum connected relay node placement* for (r, R, P, Q) (denoted by M-RNPc) if it has the minimum size among all P-RNPc for (r, R, P, Q) .

Definition 3: Let $R \geq r > 0$ be the communication ranges for RNs and SNs respectively. Let P be a set of SNs, and CL be a set of candidate locations where RNs can be placed. The *connected relay node placement problem* for (r, R, P, CL) , denoted by $\text{RNPc}(r, R, P, CL)$, seeks an M-RNPc for (r, R, P, CL) .

Theorem 1: If a set of restricted RN candidate locations is estimated then an obstacle resistant RNP strategy will result into a minimum connected relay node placement with obstacle avoidance.

Proof: Let $R \geq r > 0$ be the respective communication ranges for RNs and SNs. Let P be a set of SNs and Q be the set of RNs, and CL be a set of candidate locations where RNs can be placed without considering obstacles. Given a set F of failed SN's which are used to estimate the locations where RN's are not to be placed in case of obstacles in the deployment area. Let EL be the set of such estimated locations. So the final candidate locations for relay node placement will be the set FL i.e. $(FL = CL - EL)$. In this work our aim is to perform relay node placement over surviving SNs in order to maintain connectivity among nodes and increase lifetime of network. Formally, the surviving SNs $u \in S$ where, $S = P - F$ will be connected with RNs. As per the Definition 2 and 3, ORRNP problem can be formulated as *connected relay node placement problem* for (r, R, S, FL) , denoted by $\text{ORRNPc}(r, R, S, FL)$ which seeks an M-ORRNPc for (r, R, S, FL) .

Theorem 2: The worst case time complexity to estimate an obstacle is $(n \log n)$, where n is the number of failed sensor nodes in the network.

Proof: Obstacle estimation procedure Delaunay Triangulation is constructed in $O(n \log n)$ time. Further selection of d-simplexes, cascaded union of selected d-simplexes, and output

of boundary points is performed in $O(n)$ time. Hence the total time complexity of shape estimation is $O(n \log n)$.

Theorem 3: The worst case time complexity of ORRNP approach is $O(n^3 \log n)$ for Rectilinear SMT and $O(n^4 \log n)$ for Euclidean SMT.

Proof: When no obstacle is considered in deployment area then for n SNs Rectilinear SMT can be computed in $O(n^3 \log n)$ and Euclidean SMT in $O(n^4 \log n)$. Suppose we have n' points (failed SNs) lying in obstacle covered area. Then the candidate locations for Steiner points are reduced to $(n - n')$. This reduces the time of Rectilinear SMT generation to $O((n - n')^3 \log(n - n'))$ and Euclidean SMT generation to $O((n - n')^4 \log(n - n'))$. If the number of points (failed SNs) in obstacle covered area are too less as compared to working SNs ($n' \ll n$), then candidate locations will be almost n as $n - n' = n$ (*approximately*), if $n' \ll n$. Hence, the worst case complexity of Rectilinear SMT generation will be $O(n^3 \log n)$ and Euclidean SMT generation will be $O(n^4 \log n)$.

IV. PROPOSED APPROACH

Our proposed approach is divided into two steps:

1. Obstacle shape estimation: This part is further divided into two parts. In first part concept behind obstacle estimation is explained in brief. Second part covers mechanism behind the obstacle shape estimation approach.

A. Concept:

Based on the idea of Eddsbrunner et al. [12] shape of any obstacle is approximated. We use the location of the failed SNs to approximate the obstacle shape. The coordinates of the failed SNs is its last known GPS location. These coordinates are the input to obstacle estimator procedure which approximates shape and size of obstacles in present given deployment area.

B. Mechanism:

Any shape can be generalized by Alpha-shapes [12]. These shapes are a family of straight Line graphs. Alpha shapes can be defined as a polytypic generated over a point set P , which depends on set p and a parameter α . This parameter restricts the detail level of corresponding alpha shape. Radius $1/\alpha$ is assumed as the sensing range of any SN in case of WSNs. Fig 3. illustrates the methodology used in obstacle estimation.

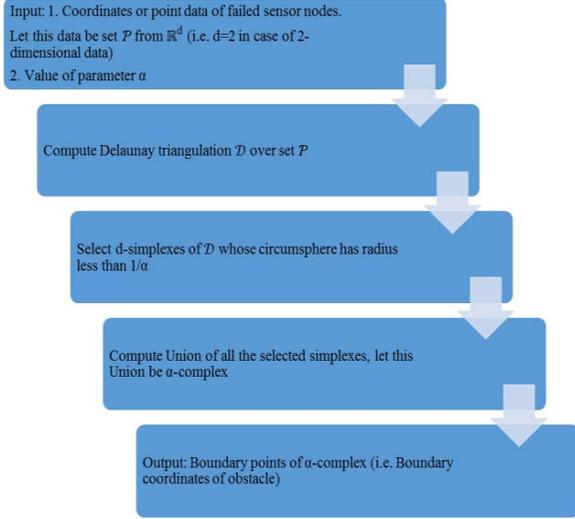


Fig 3. Obstacle estimation procedure.

The underlying space of α -complex is the area which obstacle is present. Placement of RNs must be avoided in this underlying space. As mentioned above, the parameter plays an important role in estimation of obstacle shape thus a proper input of its value is required for efficient computation and better approximation of obstacle shape. α value 0 outputs convex hull and 1 outputs Delaunay triangulation of coordinate set. Fig. 4. depicts few examples of estimated obstacles. Obstacle estimation procedure estimates the obstacle shape and area over which relay node placement is to be avoided. The output boundary coordinates of this procedure are used in ORRNP procedure.

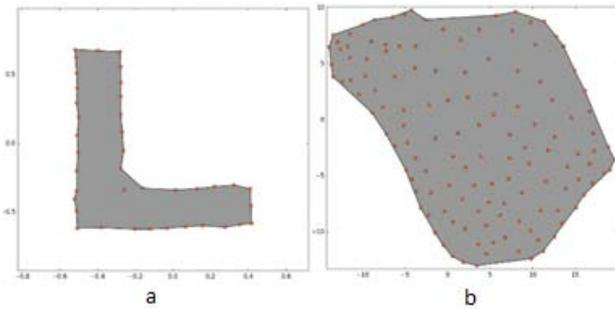


Fig 4(a, b). Examples of Obstacle types.

2. Relay Node deployment mechanism: This part presents the ORRNP strategy for estimation of RN locations when obstacles are present in deployment area.

In this work we have solved the RNP problem in constrained environment using SMT [13] approach for both rectilinear and Euclidean case. Every Steiner point has degree three and these

points are the locations representing relay nodes. SMT are further classified in two types. First is rectilinear Steiner minimum tree (RSMT) [14] in which edges are restricted to be only in horizontal or vertical direction and distance between two terminals is computed by rectilinear distance method. The second is Euclidean Steiner minimum tree (ESMT) [15]. In ESMT there is no restriction on edge direction and distance between two terminals is computed by Euclidean distance method. In Fig. 5 the pseudo code for computing RSMT and ESMT is formulated. Various other procedure are given which are being used during computation of trees. We have used simple heuristics to reduce the candidate locations set for Steiner points. Unnecessary computation is avoided by the applied heuristics and hence running time complexity of proposed approach is improved. Candidate set of locations is reduced by avoiding useless locations while computing SMT. The procedure is divided into parts which include COMPUTE-RSMT which computes rectilinear SMT similarly COMPUTE-ESMT which computes Euclidean SMT, MODIFIED-KRUSKAL part computes Minimum spanning tree with obstacle avoidance, DELTA-MST computes cost of tree when any new node is added, HANAN-GRID-SET generates a set of Hanan points which are used while computing RSMT. Pseudo code for our proposed approach has been presented in Fig 5. C_SET represent the candidate set of locations where RNs can be placed. DeltaCost variable is used to store the total cost of tree (sum of edge length) if a particular steiner point is added to it.

Initialization:

I1: Obstacles: Boundary points of α -complex shape (i.e. boundary coordinates of obstacles)
I2: Co-ordinates of working sensor nodes (i.e. nodes outside the obstacles covered area)

```

/* Computes Euclidean Minimum Steiner tree with Obstacle(s) */
COMPUTE-ESMT(I1, I2)
  C SET ← COMPUTE-C-SET(I1, I2)
  Initialize MAX_PT to a positive number
  while MAX_PT ≥ 0
    COST = 0
    Assign MAX_PT a negative value
    for each point p ∈ C SET
      DeltaCost ← DELTA-MST(I2+ SteinerPts, p)
      if DeltaCost > COST
        COST ← DeltaCost
    if MAX_PT ≥ 0
      SteinerPts = SteinerPts ∪ {MAX_PT}
    for each point p ∈ SteinerPts
      if Degree ≤ 2
        SteinerPts = SteinerPts - {p}
  FINAL ESMT ← KRUSKAL(I2 + SteinerPts)
  DISTANCE_ESMT ← ∑ Edges.distance of FINAL_ESMT
  return FINAL_ESMT
  
```

```

/* Computes Candidate set of points which can be used as Steiner Points
required for COMPUTE-ESMT */
COMPUTE-C-SET(I1, I2)
  C_SET ← {}
  ConvexHull = Convex_Hull(I2)
  for each point p ∈ Points inside ConvexHull
    if point p ∉ Points within Obstacles
      C_SET = C_SET ∪ {p}
  return C_SET

/* Computes difference between costs of two MST's one before and one
after adding SteinerPoint */
DELTA-MST(SetOfPoints, TestPoint)
  MST_ONE ← KRUSKAL-MODIFIED(SetOfPoints,
Obstacles)
  MST_TWO ← KRUSKAL-MODIFIED(SetOfPoints +
TestPoints, Obstacles)
  COST1 ← ∑ Edges.distance of MST ONE
  COST2 ← ∑ Edges.distance of MST_TWO
  return COST1 – COST2

/*Computes cost of MST; if edge intersects Obstacle(s) then distance is
set to infinity */
KRUSKAL-MODIFIED(SetOfPoints, Obstacles)
  Edges ← {}
  for each point p ∈ SetOfPoints
    for each point p' ∈ SetOfPoints
      if edge(p, p') intersects Obstacles
        Distance ← ∞
      else
        Distance ← CALCULATE-DISTANCE (p, p')
        Edges.add(p, p', Distance)
  Sort(Edges, key = Distance)
  MST ← {}
  while MST does not contains (SetOfPoints – 1) edges and
MST does not connect all SetOfPoints
    for each edge ∈ Edges
      if edge(p, p') does not create cycle
        MST = MST ∪ {edge(p, p')}
  return MST

/* Computes Rectilinear Minimum Steiner Tree with Obstacle(s) */
COMPUTE-RSMT(I1, I2)
  C SET ← HANAN-GRID-SET(I1, I2)
  Initialize MAX_PT to a positive number
  while MAX_PT ≥ 0
    COST = 0
    Assign MAX_PT a negative value
    for each point p ∈ C SET
      DeltaCost ← DELTA-MST(I1+ SteinerPts, p)
      if DeltaCost > COST
        COST ← DeltaCost
    If MAX_PT ≥ 0
      SteinerPts = SteinerPts ∪ {MAX_PT}
    for each point p ∈ SteinerPts
      if Degree ≤ 2
        SteinerPts = SteinerPts –
{p}
  FINAL_RSMT ← KRUSKAL(I2 + SteinerPts)
  DISTANCE_RSMT ← ∑ Edges.distance of FINAL_RSMT
  return FINAL_RSMT

```

```

/*Computes set of Hanan Points excluding points in Obstacle(s); which
can be used as Steiner Points in COMPUTE-RSMT */
HANAN-GRID-SET(I1, I2)
  C_SET ← {}
  HananPoints ← HANAN-GRID(I2)
  for each point p ∈ HananPoints
    if point p ∉ Points within Obstacles
      C_SET = C_SET ∪ {p}
  return C_SET

```

Fig. 5. Pseudo code of our proposed Obstacle-resistant Relay Node Placement algorithm.

V. SIMULATION RESULTS

This section presents the details of simulation environment used for investigating the performance of proposed approach ORRNP. All procedures have been coded in Python programming language using Wing IDE 101 version 5.1. The computational experiments have been executed on computer with Intel i5 II gen 64 × 2, 2.5 GHz and 4 GB RAM. The parameters α is changed according to the sensor node range while computing obstacles. Fig. 6(a) and Figure 6(b) show a case of RSMT and ESMT respectively when single obstacle is considered in deployment area. Figure 6(c) and 6(d) show a case of RSMT and ESMT respectively when multiple obstacles are considered in deployment area. In first step ORRNP finds SMT (Rectilinear or Euclidean) over set of working SNs. White points (○) represent SN and Black points (●) represent RN. Obstacles are highlighted for better understanding.

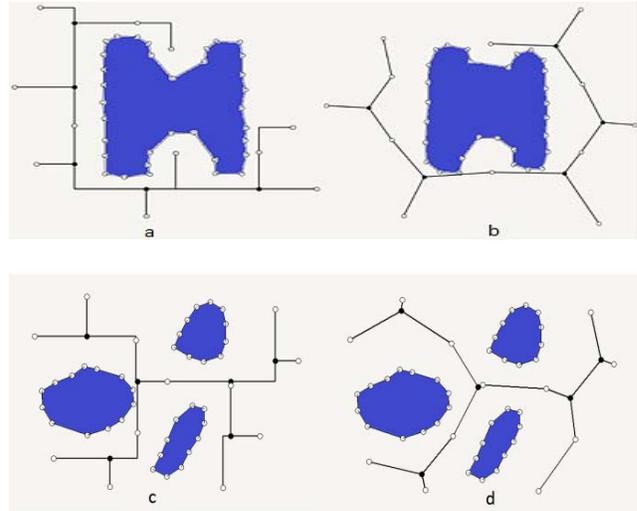


Fig. 6. a) RSMT for single obstacle b)ESMT for single obstacle c) RSMT for multiple obstacles d) ESMT for Multiple obstacles.

VI. CONCLUSION AND FUTURE SCOPE.

In this paper, the RNP in constrained environment is examined and possible solution has been formulated. The objective of this work is to perform the RNP in WSNs when the obstacles are present in deployment area. Since RNP in constrained environment is an NP-hard problem, a strategy to find a good solution in a reasonable amount of time is formulated. In future we will try to minimize the running time of ORRNP, integrate coverage factor and fault tolerance.

REFERENCES

- [1] Ranga V., Dave M. and Verma A. K., "Network Partitioning Recovery Mechanisms in WSNs: a Survey.", *Wireless Personal Communications*, vol. 72, no. 2, pp. 857-917, 2013.
- [2] Ranga V., Dave M. and Verma A. K., "Relay Node Placement for Lost Connectivity Restoration in Partitioned Wireless Sensor Networks.", *In Proceedings of International Conference on Electronics and Communication Systems (ECS 2015)*, Barcelona, Spain, pp. 170-175, 2015.
- [3] Karaki J. and Kamal A., "Routing techniques in wireless sensor networks: a survey," *Journal of IEEE Wireless Communications*, vol. 11, no. 6, pp. 6-28, 2004.
- [4] Yu M., Song J., and Mah P., "Rnindoor: A relay node deployment method for disconnected wireless sensor networks in indoor environment." *In Proceedings of 3rd International Conference on Ubiquitous and Future Networks*, Dalian, China, pp. 19-24, 2011.
- [5] Wang Y. C., Hu C. C., and Tseng Y.C., "Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks.", *In Proc. IEEE First Int'l Conf. Wireless Internet (WICON '05)*, pp. 114-121, 2005.
- [6] Verma A., Ranga V., and Angra S., "Relay node placement techniques in wireless sensor networks.", *IEEE International Conference on Green Computing and Internet of Things (ICGCIoT)*, Noida, India, pp. 1384-1389, 2015.
- [7] Sharma R., Ranga V., and Angra S., "Placement of relay nodes in wireless sensor networks.", *IEEE International Conference on Green Computing and Internet of Things (ICGCIoT)*, Noida, India, pp. 999-1004, 2015.
- [8] Ranga V. Dave M. and Verma A.K., "Relay Node Placement to Heal Partitioned Wireless Sensor Networks.", *Computers and Electrical Engineering (Elsevier)*, vol. 48, pp. 371-388, 2015.
- [9] Ranga V. Dave M. and Verma A.K., "Optimal nodes selection in wireless sensor and actor networks based on prioritized mutual exclusion approach.", *Kuwait Journal of Science*, vol. 43, no. 1, pp. 150-173, 2016.
- [10] Chang C.Y., Chang C.T., Chen Y.C., and Chang H.R., "Obstacle-Resistant Deployment Algorithms for Wireless Sensor Networks.", *IEEE Trans. Vehicular Technology*, vol. 58, no. 6, pp. 2925-2941, July 2009.
- [11] Ramer U., "An iterative procedure for the polygonal approximation of plane curves.", *Computer Vision, Graphics, and Image Processing*, vol. 1, pp. 244-256, 1972.
- [12] Edelsbrunner H., Kirkpatrick D.C., and Seidel R., "On the shape of a set of points in the plane.", *IEEE Transactions on Information Theory*, pp. 2955-2959, 1983.
- [13] Hwang F.K. and Richards D.S., "Steiner Tree Problems.", *Networks*, vol. 22, pp. 55-89, 1992.
- [14] Garey M.R. and Johnson D.S., "The rectilinear Steiner tree problem is NP-complete.", *SIAM Journal of Applied Mathematics*, vol. 31, no. 4, pp. 826-834, 1977.
- [15] Winter P. and Zachariasen M., "Euclidean Steiner minimum trees: An improved exact algorithm.", *Networks*, vol. 30, pp. 149-166, 1997.