Budget Constrained Relay Node Placement Problem for Maximal "Connectedness"

Anisha Mazumder, Chenyang Zhou, Arun Das and Arunabha Sen
School of Computing, Informatics and Decision System Engineering
Arizona State University, Tempe, Arizona 85287
Email: {anisha.mazumder, czhou24, arun.das, asen}@asu.edu

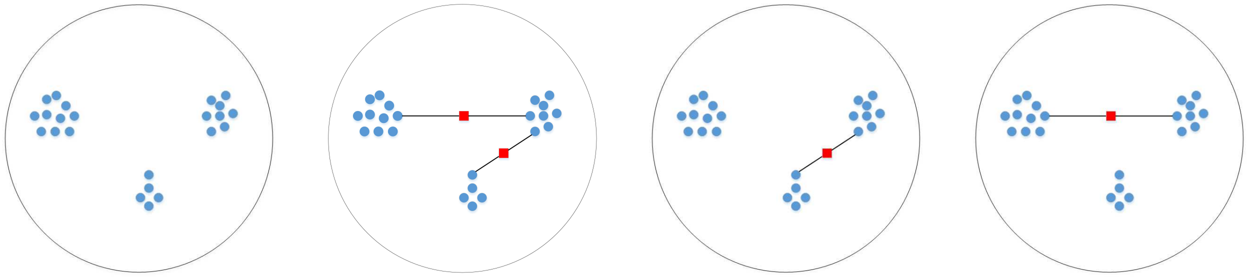
***Abstract*—The relay node placement problem in the wireless sensor network have been studied extensively in the last few years. The goal of most of these problems is to place the fewest number of relay nodes in the deployment area so that the network formed by the sensors nodes and the relay nodes is *connected*. Most of these studies are conducted for the unconstrained budget scenario, in the sense that there is an underlying assumption that no matter however many relay nodes are needed to make the network connected, they can be procured and deployed. However, in a fixed budget scenario, the expenses involved in procuring the minimum number of relay nodes to make the network connected may exceed the budget. Although in this scenario, one has to give up the idea of having a network connecting all the sensor nodes, one would still like to have a network with high level of "connectedness". In the paper we introduce two metrics for measuring "connectedness" of a disconnected graph and study the problem whose goal is to design a network with *maximal* "connectedness", subject to a fixed budget constraint. We show that both versions of the problem are NP-complete and provide heuristics for their solution. We show that the problem is *non-trivial* even when the number of sensor nodes is as few as *three*. We evaluate the performance of heuristics through simulation.**

I. Introduction

The relay node placement problem, because of its importance in wireless sensor networks, has been studied fairly extensively in the last few years [1-7]. The study of this problem is conducted in a scenario where a number of sensors (nodes) have been placed in a deployment area and often the objective is to place the fewest number of relay nodes in the deployment area such that the resulting network comprising of sensor and relay nodes is *connected*. As the deployment of relay nodes involves *cost*, it may not be possible to acquire and deploy the number of relay nodes necessary to make the entire network connected, particularly when one has to operate under a fixed budget. Although in this scenario, one has to give up the idea of having a network connecting all the sensor nodes, one would still like to have a network with high level of "*connectedness*". In this paper we introduce the notion of "*connectedness*" for a *disconnected* graph and provide two *metrics* to measure it. The first metric to measure *connectedness* of a disconnected graph is the *number of connected components* of the graph. A *lower number of connected components* in a disconnected graph is an indicator of a *higher degree of connectedness* of the graph. The second metric to measure *connectedness* of a disconnected graph is the *size of the largest connected component* of the graph. A *larger size of the largest connected component* in a disconnected graph is an indicator of a *higher degree of connectedness* of the graph. In this paper we study the problem whose goal is to design sensor networks with relay nodes to *maximize "connectedness"* subject to a fixed budget constraint. Although resource constrained version of relay node placement problems have been studied in literature [5-7], to the best of our knowledge, the problems investigated in this paper have not been studied earlier.

The problem scenario studied in this paper is depicted diagrammatically in Fig 1. By *communication range*, we refer to the upper bound on transmission range. Consider a set of twenty-three sensor nodes (shown as blue circles) deployed as shown in Fig. 1(a). Since the mathematical abstraction of the relay node placement problem corresponds to the *Geometric Steiner Tree Problem*, and the terms *Steiner Points* and *terminal points* are used in the abstraction, where the Steiner Points and terminal points correspond to the locations of the relay and sensor nodes respectively, in this paper we have used the terms "sensor nodes" and "terminal points" interchangeably. In Fig. 1(a) there are three clusters – the first one with ten terminal points, the second one with eight, while the third with five. The intra-cluster distances are within the communication range, whereas the inter-cluster ones are not. Suppose that the maximum inter cluster distance is less than twice the communication range, and as such only one relay node is sufficient for connecting any two clusters. If we have the option of placing two relay nodes (shown as red squares), then under both metrics of connectedness, the placement of relay nodes as shown in Fig. 1(b) is an optimal solution. However, if we have a budget of only one relay node, the solution shown in Fig. 1(c) is an optimal solution under budget constraint according to the first metric of connectedness. This is true as there are exactly two connected components which is the best that can be achieved with only one relay node. However, in this solution, the largest connected component has thirteen nodes and is not optimal as per the second metric. Fig. 1(d) shows the optimal placement of the relay node under budget constraint for the second metric, where the largest connected component has eighteen terminal points. It may be noted that this placement also results in an optimal solution under budget constraint according to the first metric. In this paper, we have reported our findings using these two metrics. However, we have also considered a unified

978-1-5090-3781-0/16/$31.00 ©2016 IEEE



(a) Deployment of terminal points

(b) Optimal solution with a Budget of 2 relay nodes

(c) Optimal solution for objective 1, but not for objective 2

(d) Optimal solution for both objectives 1 and 2

Fig. 1: Figure showing variation in placing relay nodes for different objectives and budget constraints

metric that measures "connectivity" of a disconnected graph by combining the two metrics discussed here. Due to lack of space, we are unable to report our findings using this unified metric in this paper.

II. Problem Formulation

As discussed earlier, the goal of this study is to enhance (or maximize) the "connectedness" of a wireless sensor network with the deployment of a limited number of relay nodes. As a first step in this direction, we formalize the notion of "connectedness" in two different ways, and accordingly, formalize two separate problems. In both problems, we are given: (i) the locations of a set of sensor nodes (terminal points) $P = \{p_1, p_2, \ldots, p_n\}$ in the Euclidean plane, (ii) the communication range R of the sensor and relay nodes, and (iii) a budget B on the number of relay nodes that can be deployed in the sensing field. From the set of points P and communication range R, we construct a graph $G = (V, E)$ in the following way. Corresponding to each point $p_i \in P$ we create a node $v_i \in V$ and two nodes v_i and v_j have an edge $e_{i,j} \in E$ if the distance between the points p_i and p_j is at most R. It may be noted that the graph $G = (V, E)$ so constructed may be *disconnected* (i.e., it might comprise of a number of *connected components*). The purpose of deploying the relay nodes is to make the *augmented graph*, $G' = (V', E')$, (comprising of sensor and relay nodes) *connected*. Suppose that the B relay nodes are deployed at points $Q = \{q_1, q_2, \ldots, q_{|B|}\}$. Corresponding to every point $q_i \in Q$ there is a node $v_i \in V' - V$ and there is an edge between v_i and a node $v_j \in V'$ if the distance between the corresponding points q_i and p_j is at most R (v_j corresponds to p_j). With unlimited budget B, obviously this goal can be achieved. However, if the budget is smaller than the minimum number of relay nodes necessary to make the graph $G' = (V', E')$ connected, this goal is unachievable. However, in this scenario, one would like to have the graph $G' = (V', E')$ with *as much connectedness as possible*. This gives rise to the "connectedness" maximization problem. The goal of creating the graph $G' = (V', E')$ with *as much connectedness as possible*, can be achieved by (i) deploying the relay nodes in a fashion that *minimizes the number of connected components of* $G' = (V', E')$, or (ii) deploying the relay nodes in a fashion that *maximizes the size of the largest connected components of* $G' = (V', E')$. We refer to (i) as *Budget Constrained Relay node Placement with Minimum Number of Connected Components* (BCRP-MNCC) problem, and (ii) as *Budget Constrained Relay node Placement for Maximizing the Largest Connected Component* (BCRP-MLCC) problem. In BCRP-MNCC, a *smaller number of connected components* is an indicator of a *higher level of connectedness* of the network. While in BCRP-MLCC, a *larger size of the largest connected component* is an indicator of a *higher level of connectedness* of the network. We formally define these two problems as follows:

Budget Constrained Relay node Placement with Minimum Number of Connected Components (*BCRP-MNCC*)
Given the locations of n sensor nodes in the Euclidean plane $P = \{p_1, p_2, \ldots, p_n\}$, positive integers R, C, and a budget B_1 on the number of available relay nodes, is it possible to find a set of $Q = \{q_1, q_2, \ldots, q_{|B_1|}\}$ points in the same plane where relay nodes can be deployed, so that the number of connected components in the graph $G' = (V', E')$ corresponding to the point set P and Q is at most C?

Budget Constrained Relay node Placement with Maximum size of Largest Connected Component (*BCRP-MLCC*)
Given the locations of n sensor nodes in the Euclidean plane $P = \{p_1, p_2, \ldots, p_n\}$, positive integer R, C, and a budget B_2 on the number of available relay nodes, is it possible to find a set of $Q = \{q_1, q_2, \ldots, q_{|B_2|}\}$ points in the same plane where relay nodes can be deployed, so that the size of the largest connected component in the graph $G' = (V', E')$ corresponding to the point set P and Q is at least C?

The authors in [8] have shown that the *Steiner Tree Problem with Minimum Number of Steiner Points* (STP-MSP) is NP-complete. As STP-MSP problem is a special case of both BCRP-MNCC and BCRP-MLCC problems, and STP-MSP is NP-complete, we can conclude that both BCRP-MNCC and BCRP-MLCC problems are NP-complete. In the following, we elaborate on this point, starting with the formal statement of the STP-MSP problem.

Steiner Tree Problem with Minimum Number of Steiner Points (*STP-MSP*): Given a set of n terminals points (location of sensor nodes) $X = \{p_1, p_2, ..., p_n\}$ in the Euclidean plane, and positive integers R and B_3, is there a tree T spanning a superset of X such that each edge in the tree has a length of no more than R and the number $C(T)$ of points other than those in X, called Steiner points is at most B_3? [8]

It may be observed that a special case of the BCRP-MNCC problem where $B_1 = B_3$ and $C = 1$ is equivalent to the STP-MSP problem. Similarly, it may be observed that a special case of the BCRP-MLCC problem where $B_2 = B_3$ and $C = n + B_2$, is equivalent to the STP-MSP problem. Since both BCRP-MNCC and BCRP-MLCC problems are generalization of the STP-MSP problem, we can conclude that both BCRP-MNCC and BCRP-MLCC problems are NP-complete.

III. Problem Solution

The budget unconstrained version of the relay node placement problem is equivalent to the STP-MSP problem discussed earlier. The authors in [8] have shown that the problem is NP-complete and provided an approximation algorithm with a performance bound of 5. A follow-up paper has since reduced the factor to 3 [9]. The approximation algorithm in [8] follows a *Minimum Spanning Tree* (MST) based approach. Although such an approach provides a constant factor approximation algorithm for the budget unconstrained version of the relay node placement problem, such an approach cannot provide a constant factor approximation algorithm for the budget constrained version of the problem as shown in the example of Fig. 2. In the figure there are three adjacent squares where length of each side is $R' + \epsilon$ and the distance from the circumcenter of the square to a corner point is R'.

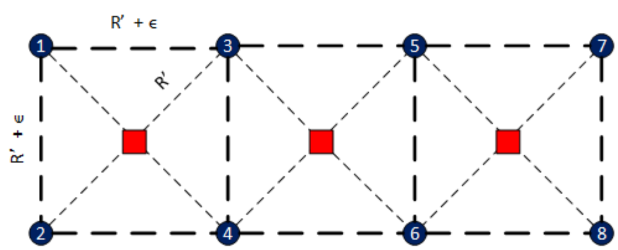


Fig. 2: Example to demonstrate that the ratio between the approximate to optimal can be $O(n)$ for any MST based approximation algorithm for BCRP-MNCC problem

In the BCRP-MNCC problem, the goal is to minimize the number of connected components subject to the budget constraint. If we only consider the square with points 1 through 4 and the budget is 1, the optimal number of connected components will be 1 by placing the relay node at the circumcenter of the square. However, in this case the MST based approach will produce 3 connected components as only two of the nodes (1, 2), (1, 3), (2. 4) or (3, 4) can be connected by a single relay node, if the location of the relay node is constrained to be on a line of the MST. Using the same argument, when the locations of the sensor nodes is points 1 through 6 and the budget is 2, the the optimal number of connected components will be 1 whereas the MST based approach will produce 4 connected components. If the locations of the sensor nodes is points 1 through 8 and the budget is 3, the optimal number of connected components will be 1 whereas the MST based approach will produce 5 connected components. In general, such a placement of sensor nodes with $n/2$ nodes on the top row and $n/2$ nodes on the bottom row as shown in Fig. 2 and a budget of $n/2-1$, whereas the optimal placement will produce 1 connected component, the MST based approach will produce $n/2+1$ components. As the the number of sensor nodes n can be arbitrarily large, the ratio between the approximate to the optimal solution of the BCRP-MNCC problem for the MST based approximation algorithm can also grow arbitrarily large.

We next show in subsection III-A that the computation of the optimal solution of the BCRP-MLCC problem *even when the number of sensor nodes is as few as three is non-trivial*. And in subsection III-B we provide heuristic solutions for both the BCRP-MNCC and BCRP-MLCC problems where the number of sensor nodes can be arbitrarily large.

A. Optimal Solution for a special case of the BCRP-MLCC

When the number of nodes is 2, i.e., $n = 2$, the BCRP-MLCC problem can be solved trivially. Consider a special case of the BCRP-MLCC problem where $n = 3$, and the distance between each of these nodes is more than the transmission range R. W.l.o.g, assume that transmission range for a relay node is 1 unit, i.e. $R = 1$ otherwise we can always divide the length of each side by R. Then, for any two nodes u, v on the two-dimensional plane, let $I_{u,v}$ be the interval formed by u, v as end points, and let $|I_{u,v}|$ be the length of the interval, the subsequent observation and lemmas follow:

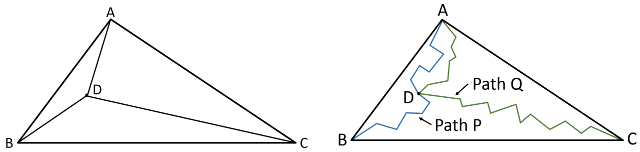
Observation 1. *If we want to make u communicate with v (in isolation w.r.t. other nodes), let the minimum number of relay nodes we need to place be $f(u, v)$, then $f(u, v) = \lceil |I_{u,v}| \rceil - 1$.*

Lemma 1. *Let u, v, x, y be four nodes on the two-dimensional plane, if $|I_{u,v}| \geq |I_{x,y}|$, then $f(u, v) \geq f(x, y)$.*

Lemma 2. *If $|I_{u,v}|$ is an integer and $|I_{u,v}| - 1 < |I_{x,y}| \leq |I_{u,v}|$, then $f(u, v) = f(x, y)$.*

Given three nodes A, B, C on the two-dimensional plane, we want to find the minimum number M of relay nodes such that A, B, C can communicate with each other. If B_2 is at least M, then the optimal solution is 3. Otherwise, the optimal solution is at most 2 and can be computed trivially. Here, we assume A, B, C are not on a straight line, otherwise, the problem can be solved easily by considering two intervals. Therefore, we consider the setting that A, B, C forms a triangle. It may be also noted that if the length of the smallest side of the triangle is at most 1, the problem becomes trivial as well. W.l.o.g, we say that the side A, B is shorter than 1, then A, B can communicate with each other directly. So we only need to consider link A, C or B, C. From Observation 1 the solution will be $\min\{f(A, C), f(B, C)\} = \min\{\lceil |I_{A,C}| \rceil - 1, \lceil |I_{B,C}| \rceil - 1\}$. Hence, we consider scenarios where all side lengths are greater than 1. Evidently, in such a scenario, we need to place at least one relay node and we should place all relay nodes within the triangle area.

Claim 1. *There exists an optimal solution which contains a relay node D, such that all the other relay nodes are located on the intervals $I_{A,D}$ and $I_{B,D}$ and $I_{C,D}$. In other words, the resulting solution looks like a star as shown in Fig. 3(a).*



(a) Figure depicting the point D. (b) Figure with paths P and Q.

Fig. 3: Constructions for proof of Claim 1

Proof. Given any optimal solution, we know the location of all relay nodes. Since A, B can communicate with each other (Fig. 3(b)), there must be a path $A - B$, path $P = (A = v_1, v_2, ..., v_n = B)$ using relay nodes as intermediate vertices. Similarly, there is an $A - C$ path $Q = (A = u_1, u_2, ..., u_n = C)$. It can be noted that there is no other relay node that is not in $P \cup Q$ as A, B, C is already connected.

We say D is the common node of P, Q, in addition, D has the largest index on P. Such a D exists since $A = v_1 = u_1$ is a candidate. After obtaining D, we divide P into two sub-paths $A - D$ and $D - B$. Since our objective is to minimize the number of relay nodes, both of these sub-paths should be intervals. We consider the same for path Q, and the resulting shape looks like a star (in some cases, the resulting shape overlaps two sides of the triangle when D is located at the same location as one of A or B or C. $\square$

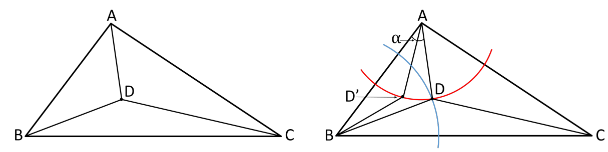
For any triangle, w.l.o.g, say (B, C) is the longest side with length L. Then, it takes at least $\Theta(\lceil L \rceil)$ time to compute the coordinates of all the relay nodes. Next we will present an algorithm that finds the minimum number of required relay nodes in $O(L^2)$ time. The main idea behind the algorithm is to consider the possible options for the optimal location of D. We see that once the location of D is fixed, the other relay nodes can be placed greedily at unit distance apart (since $R = 1$) from each other along $I_{A,D}, I_{B,D}$ and $I_{C,D}$ and we can conclude upon the required minimum number of relay nodes for this choice of location of D. We categorize the different options of location of D into three major 'Scenarios' which are further divided into different cases. For each setting, we compute the optimal location of D and the total number of relay nodes needed for that choice of D. We finally consider the location of D which minimizes the total required number of relay nodes over all categories to obtain the solution for BCRP-MLCC when $n = 3$. The three major scenarios considered are as follows:

Scenario 1: D is located inside the triangle (not on a side).
Scenario 2: D is located on side AC.
Scenario 3: D is located on either side BC or AB.
We describe Scenario 1 in details and omit the descriptions and analysis of Scenarios 2 and 3 due to lack of space.

Scenario 1: As mentioned earlier, Scenario 1 is when D is located inside the triangle (not on a side) shown in Fig. 4(a).

Claim 2. *In scenario 1, there is an optimal solution such that $|I_{C,D}|$ is an integer.*

Proof. We pick an optimal solution such that $\alpha = \angle BAD$ is the smallest. Since D is located inside the triangle, $\alpha > 0$. Let $CIR_{P,r}$ be the circle whose centre is P with radius R,

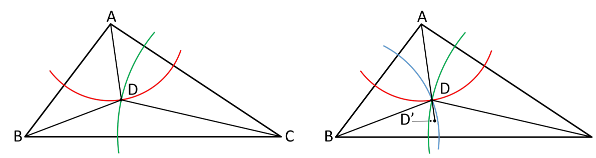


(a) Location of point D (b) Claim 2 Proof Construction

Fig. 4: Scenario 1

then D is on the circumference of $CIR_{A,|I_{A,D}|}$ as well as $CIR_{B,|I_{B,D}|}$ as shown in Fig. 4(b). Suppose $|I_{C,D}|$ is not an integer, say $|I_{C,D}| = M - \epsilon, M \in \mathbb{N}^+$. Then we can move D along circumference of $CIR_{A,|I_{A,D}|}$ a very small distance, such that $\angle BAD' < \alpha$ and $\angle D'AD \leq \min\{\alpha, \frac{\epsilon}{|I_{A,D}|}\}$. By triangular inequality , $|I_{C,D'}| < |I_{C,D}| + |I_{D,D'}| < |I_{C,D}| + |\text{DD'}| = \leq M$. According to Lemma 2, $f(C, D) = f(C, D')$. Next we consider $I_{A,D'}$. By the construction of D', $|I_{A,D}| = |I_{A,D'}|$ which implies $f(A, D) = f(A, D')$. Finally, we consider $I_{B,D'}$. Since $CIR_{A,|I_{A,D}|}$ intersects $CIR_{B,|I_{B,D}|}$ at D, D' must be within $CIR_{B,|I_{B,D}|}$, hence $|I_{B,D'}| < |I_{B,D}|$ and $f(A, D') + f(B, D') + f(C, D') < f(A, D) + f(B, D) + f(C, D)$. However, based on our choice of D and α, this is a contradiction. So, such a D' does not exist and $|I_{C,D}|$ must be an integer. $\square$

Next we show that in Scenario 1 (i) either $|I_{A,D}|$ is an integer, or (ii) one of $\angle ADC$ and $\angle ADB$ is $\frac{\pi}{2}$.



(a) Case (i) (b) Case (ii)

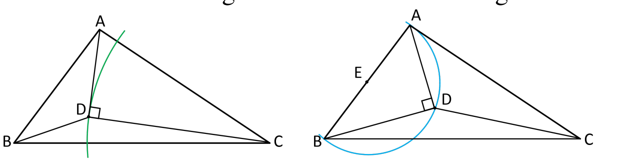
Fig. 5: Constructions for Case (i) and Case (ii) under Scenario 1

Case (i): $|I_{A,D}|$ *is integral:* In this case, as presented in Algorithm 1, we enumerate $|I_{A,D}|$ and $|I_{C,D}|$ (since both are integers), the intersection point (if there are two intersection points, we pick the one inside the triangle) of $CIR_{A,|I_{A,D}|}$ and $CIR_{C,|I_{C,D}|}$ will be the candidate of D (Fig. 5(a)). Among all candidates, the one that minimizes $f(A, D) + f(B, D) + f(C, D)$ is the final candidate.

Algorithm 1: Algorithm to compute scenario 1.(i)

for $i = 0$ to $\lfloor L \rfloor$ **do**
  for $j = 0$ to $\lfloor L \rfloor$ **do**
    Compute intersection point, say D, of $CIR_{A,i}$ and $CIR_{C,j}$ if two circle intersects.
    if the intersection point is inside triangle **then**
      Compute $f(A, D) + f(B, D) + f(C, D)$ using $f(A, D) = \lceil |I_{A,D}| \rceil - 1$ etc.
    end if
  end for
end for
Choose D that minimize $f(A, D) + f(B, D) + f(C, D)$, call it D_1.

Case (ii): $|I_{A,D}|$ *is not integral:* By the choice of D, $|I_{A,D}|$ cannot be extended. There could be only two reasons for this: either $\angle ADC = \frac{\pi}{2}$, i.e., AD is a tangent line of circle $CIR_{C,|I_{C,D}|}$; or $\angle ADB = \frac{\pi}{2}$, i.e., AD is a tangent line of circle $CIR_{B,|I_{B,D}|}$. We omit the proof of this claim due to space limitations. This gives rise to the following sub-cases:



(a) Sub-Case 1 (b) Sub-Case 2

Fig. 6: Constructions for Scenario 1, Case (ii), Sub-Cases I and II

Sub-Case I: $\angle ADC = \frac{\pi}{2}$: As presented in Algorithm 2, we can enumerate over integer values of $|I_{C,D}|$. Then we can compute a tangent AD to $CIR_{C,|I_{C,D}|}$ and get coordinates of D (Fig. 6(a)). Among all different Ds, choose the one that minimize minimizes $f(A,D) + f(B,D) + f(C,D)$ as final candidate.

Algorithm 2: Algorithm to compute scenario 1.(ii).I

for $i = 0$ to $\lfloor L \rfloor$ **do**
  Compute tangent line AD to circle $CIR_{C,i}$.
  if the intersection point is inside triangle **then**
    Compute $f(A,D) + f(B,D) + f(C,D)$.
  end if
end for
Choose D that minimize $f(A,D) + f(B,D) + f(C,D)$, call it D_2.

Sub-Case II: $\angle ADB = \frac{\pi}{2}$: Let E be the mid point of AB, then by knowledge of geometry, D lies on the circumference of $CIR_{E,\frac{|AB|}{2}}$. As presented in Algorithm 3, again we enumerate over integral values of $I_{C,D}$ and compute intersection point of $CIR_{E,\frac{|AB|}{2}}$ and $CIR_{C,|I_{C,D}|}$ (Fig. 6(b)).

Algorithm 3: Algorithm to compute scenario 1.(ii).II

for $i = 0$ to $\lfloor L \rfloor$ **do**
  Compute intersection point of $CIR_{C,i}$ and $CIR_{E,\frac{|AB|}{2}}$ where E is mid point of side AB.
  if the intersection point is inside triangle **then**
    Compute $f(A,D) + f(B,D) + f(C,D)$.
  end if
end for
Choose D that minimize $f(A,D) + f(B,D) + f(C,D)$, call it D_3.

B. Heuristic Solution for BCRP-MNCC with Arbitrary Number of Sensor Nodes

Our heuristic solution for the BCRP-MNCC problem is based on a Minimum Spanning Tree (MST) on the terminal points (sensor nodes). First we construct a complete graph on all terminal points with weights on the edges. The weight of an edge e connecting nodes v_i and v_j is equal to the Euclidean distance between the corresponding terminal points p_i and p_j divided by R, where R is the communication range, i.e., $w(e) = \lceil \frac{length(e)}{R} \rceil$. This weight $w(e)$ represents the number of relay nodes that will be needed to enable communication between the sensor nodes at the two ends of this edge. We then compute an MST on this graph. If the length of an edge of the MST is at most R, the two sensor nodes connected by this edge do not need any relay node for communication. However, if the length of an edge of the MST is greater than R, some relay nodes will be needed for communication between the sensor nodes connected by this edge. We place the relay nodes on the MST edge (i.e., the line connecting the terminal points) and the number of relay nodes needed to enable communication between two sensor nodes will be equal to $w(e)$.

If the budget on the number of available relay nodes is sufficient, i.e., if $\sum_{e \in E(T')} w(e) \leq B_1$, the number of connected component is one and we directly output the solution. Otherwise, we are short of relay nodes and we successively remove some of the edges of T' till such time that the number of required relay nodes becomes less than or equal to the budget. It may be noted, removal of one edge from the MST increases the number of connected components by exactly one. We follow a greedy approach for edge removal sequence in that at every stage of the removal process, we remove the highest weighted edge, breaking ties arbitrarily (Algorithm 4).

Algorithm 4: Heuristic for BCRP-MNCC problem

Create an MST T' on the set of given terminal points P.
Assign each edge e of T' a weight of $w(e) = \lceil \frac{length(e)}{R} \rceil - 1$
while $\sum_{e \in E(T')} w(e) > B_1$ **do**
  Remove the edge that has the maximum weight from T'; breaking ties arbitrarily
end while
Return the resulting forest obtained from T'

C. Heuristic Solution for BCRP-MLCC with Arbitrary Number of Sensor Nodes

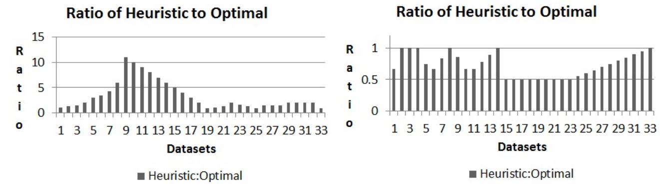
Our heuristic for the BCRP-MLCC is based on the k-MST problem, where one is given an undirected graph G with non-negative costs $c(e)$ for the edges $e \in E(G)$ and an integer k, and the problem is to find the minimum-cost tree in G that spans at least k vertices. Computation of k-MST is a well studied problem. [10] and [11] present $1+\epsilon$ approximate solutions for the k-MST problem. Our heuristic (Algorithm 5) for the BCRP-MLCC computes k-MST with decreasing value of k starting with $k = n$, where n is the number of terminal nodes. Once the k-MST is computed, our algorithm computes the minimum number of relay nodes that will be necessary to make the k-MST connected by using the same technique as in the BCRP-MLCC problem. If this number does not exceed the budget, our procedure stops and outputs the nodes of the k-MST as the largest connected component. Otherwise it computes k-MST once again with the value of k decremented by one and then checks if the number of relay nodes needed is within the specified budget.

Algorithm 5: Heuristic for solving BCRP-MLCC problem

for $k \in n\ to\ 2$ **do**
    Create an approximate k-MST T' on the set of given target points P
    To each edge e of T', assign the weight $w(e) = \lceil \frac{length(e)}{R} \rceil - 1$
    if $\sum_{e \in E(T')} w(e) \leq B_2$ **then**
        Return T' as the solution of BCRP-MLCC
    end if
end for
Return any arbitrary terminal point as solution

IV. Experimental Results

In this section, we present the results of our experimental evaluations of Algorithms 4 and 5. To compute the MST for BCRP-MNCC, we use Prim's algorithm and for k-MST for BCRP-MLCC, we use algorithm presented in [12]. In order to evaluate the performance of the heuristics for BCRP-MNCC and BCRP-MLCC problems presented in Algorithms 4 and 5, we need to know both the approximate and the optimal solution for the problem instances. Whereas, approximate (heuristic) solution to the problem instances can be obtained by running Algorithms 4 and 5, optimal solution to the problem instances is not obvious using Integer Linear Programming (which is often used in similar problems scenarios) as the placement of a relay node can be at any point in the deployment area and the number of such points are infinite. To overcome this constraint, we created data sets by placing the sensor nodes at specific locations in the deployment area so that we can compute the optimal solution for a specified budget easily. We manually created 33 datasets, each with 20 sensor nodes and a fixed communication range, varying the (i) the sensor node deployment pattern and the (ii) relay node budget, in a way that we know the optimal solution for these problem instances. The ratio between the heuristic to optimal solution for the BCRP-MNCC and BCRP-MLCC problems are shown in Fig. 7. On the X-axis of Fig. 7 we have data sets 1 through 33 and on the Y-axis have the ratio of the heuristic to the optimal solution for problem instance (i.e., a specific data set). It may be observed that the ratio between heuristic to optimal was never lower than 0.5 for the BCRP-MLCC problem but for the BCRP-MNCC problem this rato was as large as 11. As we have observed earlier in section III, an MST based solution to the BCRP-MNCC problem can perform poorly, if the sensor nodes have some specific (bad) deployment pattern. The poor performance of the BCRP-MNCC heuristic for some problem instances can be explained by this observation.



(a) Plot of experimental results for BCRP-MNCC problem

(b) Plot of experimental results for BCRP-MLCC problem

Fig. 7: Experimental results plotting the ratio of the heuristic to the optimal solutions for different datasets for the BCRP-MNCC and the BCRP-MLCC problems.

V. Conclusion

In this paper, we have studied the relay node placement problem under budget constraint using two different metrics. We prove that the problems using both the metrics are NP-complete and provide heuristic solutions for them. We report our experimental results on synthetic data sets. This study has led us to the development of the notion of "connectivity of a disconnected graph" or "disconnectivity", which we believe opens up a new area of research.

References

[1] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *Computers, IEEE Transactions on*, vol. 56, no. 1, pp. 134–138, 2007.

[2] S. Khuller and B. Raghavachari, "Improved approximation algorithms for uniform connectivity problems," in *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*. ACM, 1995, pp. 1–10.

[3] W. Zhang, G. Xue, and S. Misra, "Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 1649–1657.

[4] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 5, pp. 643–656, 2010.

[5] S. Misra, S. D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks: Formulation and approximations," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 2, pp. 434–447, 2010.

[6] J.-Y. Chang and Y.-W. Chen, "A cluster-based relay station deployment scheme for multi-hop relay networks," *Communications and Networks, Journal of*, vol. 17, no. 1, pp. 84–92, 2015.

[7] P. Li, C. Huang, and Q. Liu, "Bcdp: Budget constrained and delay-bounded placement for hybrid roadside units in vehicular ad hoc networks," *Sensors*, vol. 14, no. 12, pp. 22 564–22 594, 2014.

[8] G.-H. Lin and G. Xue, "Steiner tree problem with minimum number of steiner points and bounded edge-length," *Information Processing Letters*, vol. 69, no. 2, pp. 53–57, 1999.

[9] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, and G. Xue, "Approximations for steiner trees with minimum number of steiner points," *Journal of Global Optimization*, vol. 18, no. 1, pp. 17–33, 2000.

[10] J. S. Mitchell, "Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1298–1309, 1999.

[11] S. Arora, "Polynomial time approximation schemes for euclidean tsp and other geometric problems," in *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*. IEEE, 1996, pp. 2–11.

[12] F. A. Chudak, T. Roughgarden, and D. P. Williamson, "Approximate k-msts and k-steiner trees via the primal-dual method and lagrangean relaxation," *Mathematical Programming*, vol. 100, no. 2, pp. 411–421, 2004.