IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 1, 2022, accepted January 22, 2022, date of publication January 28, 2022, date of current version February 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3147488

Relay Placement Algorithms for IoT Connectivity and Coverage in an Outdoor Heterogeneous Propagation Environment

NIHESH RATHOD AND RAJESH SUNDARESAN, (Senior Member, IEEE)
Robert Bosch Centre for Cyber-Physical Systems, Department of Electrical Communication Engineering, Indian Institute of Science (IISc), Bengaluru 560012, India

Corresponding author: Nihesh Rathod (niheshkumar@iisc.ac.in)

This work was supported in part by the Science and Engineering Research Board, Department of Science and Technology, Government of India; and in part by the Robert Bosch Centre for Cyber-Physical Systems, Indian Institute of Science, Bengaluru. The work of Nihesh Rathod was supported by a fellowship grant from the Centre for Networked Intelligence (a Cisco CSR initiative) of the Indian Institute of Science, Bengaluru.

ABSTRACT A vast majority of the Internet of Things (IoT) devices will be connected in a topology where the edge-devices push data to a local gateway, which forwards the data to a cloud for further processing. In sizeable outdoor deployment regions, the edge-devices may experience poor connectivity due to their distant locations and limited transmission power. Repeaters or relays must be placed at a few locations to ensure reliable connectivity to either a gateway or another node in the network. A big challenge in achieving reliable connectivity and coverage is the outdoor propagation environment being heterogeneous. Engineers often deploy networks based on resource-intensive field visits, detailed surveys, measurements, initial test deployments, followed by fine-tuning. For scalability to large scale IoT deployments, automated network planning tools are essential. Such tools should predict connectivity based on the edge-device locations using available Geographical Information System (GIS) data, identify the need for relays/repeaters, and, if needed, suggest the number of relays needed with their locations. Furthermore, such tools should also be extended to suggest the minimum number and locations of base stations that maximise coverage. In this paper, we propose an automated network deployment framework using a black box received signal strength estimation oracle that provides signal strength estimates between candidate pairs of transceiver locations in a heterogeneous deployment region. Our proposed methodology uses either Ant Colony Optimisation (ACO) or Differential Evolution (DE) to identify the number and locations of relays for meeting specified quality of service constraints. We discuss adaptations of our techniques to handle scenarios with multiple gateways. Further, we show the effectiveness of these algorithms to find suitable candidate base station locations to provide coverage in a heterogeneous propagation environment that meets the specified quality of service constraints. We then demonstrate the effectiveness of our algorithms in two deployment regions.

INDEX TERMS Coverage, GIS, heterogeneous propagation environment, Internet of Things, IoT, RF propagation tool, RSSI, sub giga hertz, sub-GHz.

I. INTRODUCTION

Enabling automation in outdoor Internet of Things (IoT) network deployment is crucial for the expansion of IoT systems at the currently projected scales. Traditional network deployment strategies involved time- and engineering-resource consuming field surveys for estimating network coverage in the deployment region. Instead, if we could predict coverage in the region, based on prior knowledge of the terrain, we can save not only valuable engineering resources but also enable rapid deployment. In this paper, we highlight many interesting problems and solutions in creating workflows for automating network deployment.

Given a collection of data sources, i.e., edge-devices with data to push into the IoT network, and a destination node (a gateway to a cloud computing and storage platform), all of which are located on a Geographical Information System (GIS), the connectivity problem is to identify suitable relay

The associate editor coordinating the review of this manuscript and approving it for publication was Zhipeng Cai.

locations that can help transport data from the IoT edge-devices to the gateway by meeting certain specified Quality of Service (QoS) constraints.

We divide this problem into two subproblems.

- In the first subproblem, we predict the link quality between an arbitrary pair of points that comprise a potential transmitter-receiver pair. A solution to this subproblem is typically difficult because outdoor environments are often *heterogeneous*: carrier pathways traverse different propagation environments having different path loss exponents, shadowing parameters, fading parameters, etc., and this renders their estimation difficult.
- In the second subproblem, we deploy the network for either connectivity, as described above, or coverage, as we shall see later in the paper. Connectivity or coverage involves the determination of a suitable number of relays or base stations, their locations, their transmit powers, etc., for meeting specified QoS requirements.

One approach to solve the first problem was presented in Rathod and Sundaresan [2], [3] which provided a coverage estimation tool that could handle heterogeneity in the propagation environments. Rathod and Sundaresan [2], [3] reported extensive measurements in example environments and proposed a 'library' of propagation models. They then processed the GIS data for the deployment region and partitioned it into subregions of locally homogeneous propagation conditions. Next, they mapped each of these to one of the propagation models in the library. They then used a composite signal propagation model to predict the received signal strength indication (RSSI) between any pair of potential transmitter and receiver locations. The coverage estimation tool of [2], [3] has sufficient flexibility to be used as a black-box or an oracle to predict the RSSI between an arbitrary pair of points located in the deployment region.

In this paper, we focus on the second subproblem, which includes network deployment for connectivity and network deployment for coverage. Our contributions are as follows.

- We first formulate and address the problem of relay placement for *connectivity*. Formally, given a set of IoT edge-device locations, a gateway location, and a minimum RSSI threshold R, the connectivity problem is the following:

  Find the minimum number of relays and their locations so that there is a *(spanning) tree* with all links having RSSI at least R.

  The above relay placement problem, to meet QoS requirements, is likely to be computationally hard because it is a version of the computationally hard Steiner tree problem [4]. Furthermore, as we shall soon see, the geometric structure is complex because the link cost between an arbitrary pair of nodes depends on the link quality for wireless transmission in the associated *heterogeneous* environment. In particular, the geometric structure is more complicated than the well-studied Euclidean case.
- We propose two biology-inspired heuristics to solve this connectivity problem in the single gateway setting: the Ant Colony Optimisation (ACO) algorithm and the Differential Evolution (DE) genetic algorithm. However, both algorithms do not directly apply to our settings and require adaption to handle our objective.
- The above development is for a single gateway. We next consider a variation with multiple gateways. The modified objective of the network deployment problem for connectivity with multiple gateways is to make each IoT edge-device connect to *any one* of the gateways using the minimum number of relays, suitably located, so that the resulting "forest" has RSSI at least R on each link. Again we adapt the ACO and the DE algorithms to handle the modified objective.
- We then study the problem of network deployment for providing *coverage* in a heterogeneous environment with the minimum number of additional nodes, which we now call base stations instead of relays. We demonstrate the suitability of the DE algorithm for this problem.
- We explain our ideas, algorithms, and results by running them for the Indian Institute of Science campus (IISc campus). We study both connectivity and coverage. One reason for using the IISc campus is that, despite its relatively small size (roughly 2km-by-2km), it offers diverse propagation environments that are representative of a big city. Additionally, we explore coverage for Kakinada, the sixth-largest city in the Indian state of Andhra Pradesh.

The rest of the paper is organised as follows. In Section II, we provide a brief overview of the existing literature and the off-the-shelf commercial solutions. In Section III we introduce the relay placement problem, describe the Ant Colony Optimisation (ACO) algorithm, explain how ACO is adapted to find acceptable solutions to the NP-hard Steiner tree problem [4]. We also show some example outcomes. We then turn to the relay placement problem in a heterogeneous region with one gateway, and show how ACO can be adapted to solve it. We end the section by showing the outcomes of our adapted ACO algorithm for the IISc campus. In Section IV, we deal with the Differential Evolution (DE) algorithm and follow the same sequence of developments as in Section III. In Section VI, we extend the algorithms to address the relay placement problem with multiple gateways. We first use the ACO and DE algorithms following a local divide-and-conquer strategy. We then explain shortcomings of the local approach, then propose two algorithms to overcome the shortcomings, and compare them with each other. In the next Section VII, we introduce the coverage problem, highlight the effectiveness of the Differential Evolution (DE) algorithm in solving the coverage problem and showcase the outcomes of the DE algorithm. We end the paper with some concluding remarks. Unlike prior relay placement works (e.g., [5], [6]), the main feature of this work is the ability to handle heterogeneous environments in conjunction with the coverage tool of [3]. State-of-the-art results of [7], which are more applicable to problems arising from homogeneous propagation environments, are therefore not directly applicable to our setting of the heterogeneous propagation environment.

II. RELATED WORKS

Algorithms for relay placement is an active area of research in computer networks, Internet of Things, and wireless communications. Prior works tackled this multi-faceted problem to meet specific QoS requirements in different ways, which we now highlight. Nikolov *et al.* [8] jointly optimised the relay locations and the resulting traffic through the network in order to minimise the number of packet retransmissions. Perez *et al.* [9] proposed a hybrid evolutionary algorithm for the simultaneous optimisation of the number of relays and the energy dissipation in wireless sensor networks. Efrat *et al.* [10] showed that the *one-tier* version of relay placement problem does not have any polynomial-time approximation scheme (PTAS), given $P \neq NP$. They then presented a $3.11-$approximation algorithm for the *one-tier* version and a PTAS for the *two-tier* version. Minelli *et al.* [11] addressed the problem of placing relays in a cellular network setting with the goal of maximising the cell capacity. Here, they used a dedicated Simulated Annealing (SA) algorithm to search for an optimal solution. Li *et al.* [12] introduced the notion of 'balanced data paths' to the sink to extend the lifetime of the sensor network using a Voronoi-based placement algorithm. Lin *et al.* [13] translated the joint problem of relay placement and bandwidth allocation to an integer linear program and solved it using IBM's CPLEX tool. Al-Turjman *et al.* [14] studied the relay placement problem in a federated setting to connect disjointed sectors while maintaining cost constraints. Lately, algorithms like the Jarvis March approach [15], moth flame optimiser algorithm, interior search algorithm, and bat algorithm [16] have also been used to solve the relay placement problem. All of these works involve Euclidean costs and do not handle the complexity arising from a heterogeneous propagation environment. Thus, the distinguishing feature of this paper is our handling of heterogeneity in the propagation environment.

The network coverage problem has also been actively researched in the wireless sensor network context. Howard *et al.* [17] tackled the problem of deploying a mobile sensor network in an unknown environment. The proposed to place the base stations in such a way that each node is repelled by both obstacles and by the other nodes, so that the base stations are spread throughout the environment for better coverage. Moysen *et al.* [18] provided a data-driven machine learning (ML) framework to find the best locations of base stations for a microcell deployment. Quintão *et al.* [19] proposed evolutionary algorithms to solve the Dynamic Coverage Problem (DCP). Mahboubi *et al.* [20] used a multiplicative weighted Voronoi diagram to discover coverage holes and moved sensors to minimise the uncovered or vacant regions in the target field. Cardei and Wu [21] provided a survey of the field of energy-efficient coverage problems in the context of static wireless sensor networks. They presented coverage formulations, their assumptions, and provided an overview of some proposed solutions. Recently, Tossa *et al.* [22] used a heuristic genetic algorithm to cover a two-dimensional Euclidean area with a given number of sensors and thus found suitable placements for good network coverage. Again, we need to go a little further than these works to handle heterogeneous propagation environments, which we do in this paper.

Heterogeneous networks (HetNets), different from heterogeneous propagation conditions, have also been studied. Shin and Zain [23] maximised cellular coverage probability in a heterogeneous network by placing pico cell towers where the coverage is poor. Gazda *et al.* [24] used novel models for pedestrian and vehicular UE and employed a self-learning algorithm for optimisation of HetNet deployment using self-organising maps (SOMs). Li *et al.* [25] used Gibbs sampling based optimisation for the deployment of small cells in 3G Heterogeneous Networks. Again, we need to go a little further than these works to handle *heterogeneous propagation environments*, which we do in this paper.

There exist many network planning tools, both open source and commercial, that aim to make the process of network deployment efficient. See for example Götz [26], Teoco RAN Solutions [27], Intermap [28] and Wireless Insite [29], [30]. The problem that these tools are trying to solve is different from those articulated in our problem above. These tools try to predict the link quality or coverage area for a given transmitter location based on a variety of models such as the Longley-Rice model [31]–[33], the Edwards-Durkin model [34], [35], the Okumura model [36], the Hata model [37], the COST-231 model [38], etc. Our work instead tries to address the core problem of finding the relay and base station locations in an automated way to meet certain QoS requirements. These QoS can be estimated via the above-reported tools if the deployment area is homogeneous. However, our outdoor IoT deployment region is in general heterogeneous. To handle this heterogeneity, the aforementioned traditional models do not suffice, and we resort to using the prediction methodology proposed in [3].

III. ANT COLONY OPTIMISATION FOR RELAY PLACEMENT WITH SINGLE GATEWAY

Consider the following network deployment problem in a given deployment region. We are given the GIS data for the deployment region, the location of a single gateway, and the locations of all the IoT edge-devices (which we call transmitters). We must connect these transmitters to the single gateway, either directly or through other transmitters or through newly introduced relays, in such a way that every link in the path to the gateway has received signal strength indication (RSSI) of at least R.

Given any pair of nodes in the deployment region as input, the RSSI computing algorithm in [3] acts like an oracle and returns the average RSSI between the pair of nodes. If each transmitter has direct connectivity (RSSI exceeds R) or connectivity via other transmitters to the gateway (RSSI exceeds R in every link in the path), then no relay nodes are required, and can consider the network deployed.

If even one node is unable to reach the gateway, we will need relays. However, we must minimise the number of relays because they involve extra hardware, increased maintenance, and therefore higher cost. Identification of the minimum number of relays for connectivity is then related to the well-known Steiner tree problem. If the propagation environment is homogeneous, then the problem reduces to the Euclidean Steiner Tree Problem (Euclidean STP), which is defined as follows:

STP: Given n points in the plane $\{x_1, x_2, \ldots, x_n\}$, connect the points by line segments of minimum total length in such a way that any two points in the set may be connected either directly by a line segment, or indirectly via line segments through other points in the set, or via line segments passing through other points in the set and other new points that may be introduced for enabling the connectivity.

The resulting graph with the new points is a minimal spanning tree (after including the new points). It is well-known that for a general n, STP is NP-hard [4]. Given this computational intractability, we next describe an ant colony optimisation based meta-heuristic algorithm to find an acceptable solution. We first introduce ACO, then describe its use in a homogeneous propagation environment, and then discuss its use in a heterogeneous propagation environment.

A. ANT COLONY OPTIMISATION (ACO)

In the natural world, ants often start their search for food by wandering off, initially, in random directions. Upon finding a food source, they return to their colony, leaving the trail of a substance called "pheromone". This attracts other ants and encourages them to explore along the existing pheromone trails rather than exploration at random. As more ants find the food source, the pheromone trails to the food source get reinforced.

Pheromone trails evaporate with time, and this decreases the attractiveness of longer paths. Indeed, the more the time taken by an ant to travel on a particular path and return, the more the evaporation of pheromone on that path. On the other hand, a shorter path has lesser evaporation initially, is therefore discovered more easily and travelled more frequently. This leads to greater pheromone accumulation and a reinforcement of the shorter paths. Evaporation of the pheromone is thus the key feature that enables the discovery and reinforcement of shorter paths. Indeed, if the pheromone evaporation did not take place, then ants may get locked to the trail blazed by the first ant, thereby constraining the discovery of alternate shorter and better paths to the food source.

Ant colony optimisation (ACO) algorithm is a meta-heuristic algorithm inspired by this natural behaviour of ants. ACO is known to produce acceptable solutions within a reasonable time for some NP-hard problem instances, such as the Travelling Salesman Problem. Further, due to its iterative nature, ACO can dynamically adapt to changes in the graph structure. Due to these attractive properties, we choose ACO to solve our problem. In the rest of this section, we describe our adaptation of ACO to first solve the Euclidean STP problem and then to solve the relay placement problem in a heterogeneous environment.

B. ACO FOR THE EUCLIDEAN STEINER TREE PROBLEM

Readers familiar with the ACO for Euclidean TSP may skip this subsection, which is provided only for completeness. The ACO metaheuristic optimises a function of several variables. It takes as input the variables, their ranges, and a method to evaluate the function. The ACO also takes an initial guess as an input. Further, it has a few algorithmic parameters such as the number of ants, the maximum number of iterations, the error tolerance values, etc. We keep them fixed to default values in our experiments, and do not discuss their impact on the output of the algorithm.

Our use of ACO is as follows. We first create a function that assesses the benefit of one new relay node ($k = 1$) at location (d_{ix}, d_{iy}), as a function of this location. To evaluate this function, we form a minimum spanning tree (MST) on the graph with nodes $(d_{ix}, d_{iy}) \cup \{x_1, x_2, \ldots, x_n\}$ and link costs between a pair of points given by the Euclidean distance between the two points. We then add the link costs of all the links in this MST, and return it as the function value at the point (d_{ix}, d_{iy}). ACO takes this function and a random initial point, and tries to minimise it across (d_{ix}, d_{iy}). After executing its iterations, when either successive evaluations were within the specified tolerance or the maximum number of iterations was reached, ACO will output a possibly better location (d'_{ix}, d'_{iy}). We then restart the algorithm, but this time with two ($k = 2$) new nodes placed at random. We then continue to increase the number of extra nodes k until we reach $n - 2$, a known upper limit on the extra nodes for the Euclidean STP [39]. We then pick the best choice of k, which is the choice that yields the lowest MST weight. This approach may need multiple random restarts for each k because the iterative algorithm may settle down at a local minimum.

Fig. 1 shows ACO outcomes on Euclidean STP for transmitters placed at the vertices of regular polygons. The n transmitters were kept at vertices of different polygons. Case numbers printed next to the solutions are referenced in the discussion. The choice of regular polygons allows us to validate our implementation outcomes with those in the literature [39].

As shown in Fig. 1, in the triangle with all the angles less than 120° ($n = 3$, case 1), the ACO algorithm solves the STP with $k = n - 2 = 1$ Steiner node. Further, the location of the Steiner node is at the centroid of the triangle, which is intuitive. For a square ($n = 4$, case 3), a rectangle ($n = 4$, case 4), a parallelogram ($n = 4$, case 5), and a trapezoid ($n = 4$, case 6), the ACO output matches with the best possible solution described in the literature [39]. For $n = 4$, case 6, not only does it stop at $k = 2$, but it also gives the correct locations of the Steiner nodes.

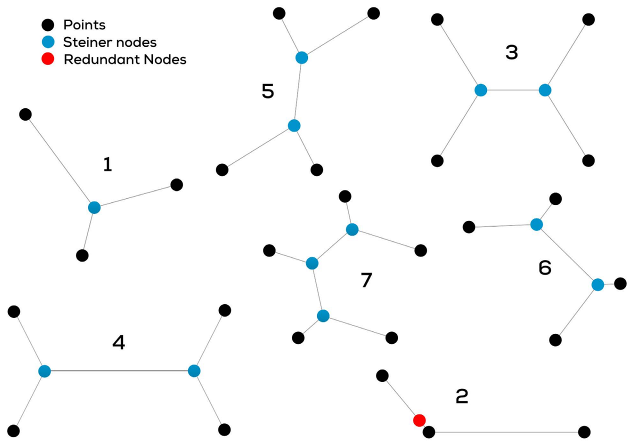

FIGURE 1. Solutions to Steiner tree problems for regular polygons.

For a pentagon ($n = 5$, case 7), the solution was obtained with $k = 3$. For Case 2, a triangle with one of the angles larger than 120° ($n = 3$), the solution has a very interesting interpretation: ACO outputs a Steiner node on the line connecting the triangle corners; this is marked as a red circle. It means that the solution to STP is to connect three points directly. There is no other better way to form the MST in this case. Interestingly, we did not configure our algorithm to treat the triangles in cases 1 and 2 differently based on the angles. The ACO algorithm identified these correct solutions without any special considerations. This highlights the robustness of the algorithm for different configurations of n points. Note that the objective function involves Euclidean distances with exploitable properties such as the triangle inequality. In the next subsection, we adapt this algorithm and highlight our approach to solve the relay placement problem with arbitrary pairwise link costs arising from propagation in a heterogeneous environment.

C. ACO FOR SOLVING THE RELAY PLACEMENT PROBLEM IN A HETEROGENEOUS REGION

We now come to the use of ACO for relay placement in a heterogeneous region. We use a fixed link cost function f that maps a link RSSI to a link cost $f(\text{RSSI})$, i.e., if i, j are a pair of nodes and $\text{RSSI}(i,j)$ is the received signal strength between the transmitter-receiver pair located at i and j, respectively, then the link cost w_{ij} is given by $w_{ij} = f(RSSI(i,j))$. We take

$$f(\text{RSSI}) = \begin{cases} -\dfrac{\text{RSSI}^{1-\alpha} - 1}{1-\alpha} & \alpha \geq 0.\alpha \neq 1 \\ -\log(\text{RSSI}) & \alpha = 1 \end{cases}$$

with a suitable parameter $\alpha \geq 0$ and with RSSI in linear scale. We will soon discuss the choice of the parameter α. The function f is the negative of the so-called generalised logarithm [40] with parameter $\alpha \geq 0$. Observe that $f(\text{RSSI}) = 1 - \text{RSSI}$ for $\alpha = 0$ and $f(\text{RSSI}) = -\log(\text{RSSI})$ for $\alpha = 1$. We will shortly discuss our choice of f and α. For now observe that it is a decreasing function of RSSI.

Note that $\text{RSSI}(i,j)$ now depends on the propagation environments in which nodes i and j are located and on the different propagation environments traversed by a path from i to j due to heterogeneity. We use the procedure of Rathod *et al.* [3] as an oracle to provide us with the RSSI between a given pair of nodes.

The problem can then be stated as follows:

Problem: Given a heterogeneous deployment region, an RSSI threshold R, n transmitter locations $\{x_1, x_2, \ldots, x_n\}$, and a gateway or aggregating location $\{y\}$ in the heterogeneous region, find the minimum number of relays k and their locations $\{d_1, d_2, \ldots, d_k\}$ so that each link e of the resulting Minimum Spanning Tree on $S = \{y\} \cup \{x_1, x_2, \ldots, x_n\} \cup \{d_1, d_2, \ldots, d_k\}$, with link costs coming from the function f that takes link RSSIs to link costs, has RSSI $\geq R$ on each link, that is,

$$\min_{\substack{(i,j)\in E(MST(S)) \\ i\neq j}} \text{RSSI}(i,j) \geq R, \tag{1}$$

where $E(MST(S))$ is the set of links of the graph $MST(S)$, a minimum cost spanning tree with the cost of each link (i,j) being $w_{ij} = f(\text{RSSI}(i,j))$.

If $f(\text{RSSI}) = 1 - \text{RSSI}$, obtained by setting $\alpha = 0$, then the $MST(S)$ attempts to get a tree with the highest possible RSSI-sum across the links in the tree. We then require that this tree has RSSI at least R on every link.

Let us now discuss the choice of the α parameter defining the function f. Observe that the following is a more ideal objective:

$$\max_{T} \min_{\substack{(i,j)\in E(T) \\ i\neq j}} \text{RSSI}(i,j) \geq R,$$

where the first maximum over T is across all trees, i.e., it suffices if there is some tree T, all of whose links have RSSI at least R. But the number of trees is $|V|^{|V|-2}$, where $|V| = n + k + 1$ is the number of nodes in the graph, by Cayley's formula, and one needs to look at the tree with the highest minimum RSSI among these superexponentially large number of trees. If we can turn the max-min objective over trees into a suitable min sum-cost objective over trees, we can then apply Prim's MST algorithm. This is exactly what is enabled by choosing f with a large α. As $\alpha \to \infty$, the min sum-cost tree approaches the max min-RSSI tree. So our heuristic in (1) is to fix a large α, solve the min sum-cost problem using Prim's algorithm, and then demand the additional condition that RSSI $\geq R$ on every link of the MST.

We are now ready to apply the ACO algorithm. See Fig. 2 for a flow diagram of our approach. As already mentioned in Section III-B, ACO is an iterative algorithm, and the iterative step is highlighted in red in Fig. 2. The function indicated "Cost$(\cdots)$" constructs the Minimum Spanning Tree on $\{y\} \cup \{x_1, x_2, \ldots, x_n\} \cup \{d_1, d_2, \ldots, d_k\}$ in each iteration and outputs the negative of the minimum RSSI (across links) on the MST, i.e., the negative of the left-hand side of (1). ACO tries to minimise this value (which is the same as maximising the minimum RSSI). Only when the algorithm converges or when the maximum number of iterations is reached do we check whether each link has the minimum required RSSI R.

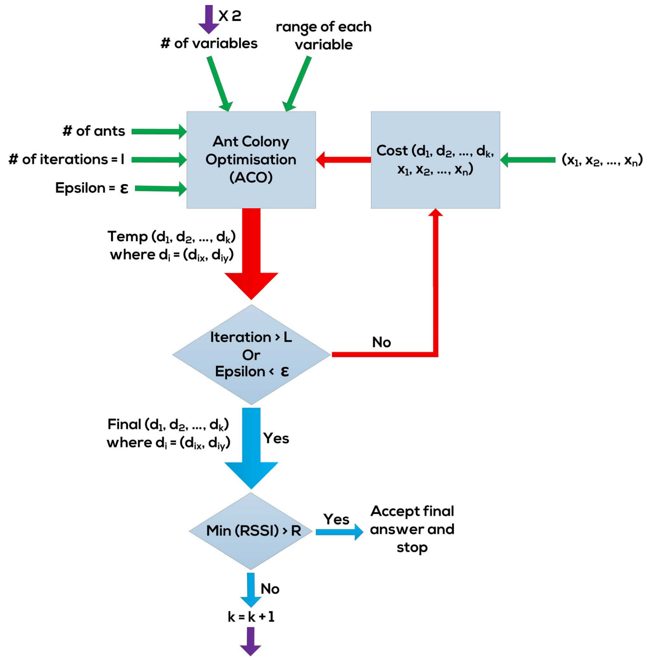

FIGURE 2. **Flow diagram of the ACO algorithm.**

If all the links have at least this minimum RSSI threshold R, then we accept the solution. If the minimum RSSI for the Steiner tree with a given number of relays is not above R, we increase the number of relays by one and start over.

We next chose several adversarial instances and conducted stress-tests on our algorithms. We set the value of parameter $\alpha = 0$, i.e. $MST(S)$ finds the tree with the highest possible RSSI-sum across the links in the tree. We set $R = -110$ dBm. As an example, in Fig. 3, we kept our transmitters and the gateway at the four corners of the image, each marked as "$+$". Any one of the four locations may be taken as the single gateway to which the other transmitters must connect. The transmitters and gateway were kept so far apart that direct communication between any pair was impossible. Then we let our algorithm suggest the relay locations, after taking heterogeneity of the propagation environment into account. Fig. 3 shows two outcomes of the experiment. The suggested relay locations are marked as red circles. In both outcomes, the number of relays used for network connectivity is six, with both outcomes meeting the minimum RSSI constraint. The locations suggested by the two outcomes are different due to the presence of several local minima (several relay location configurations) for the relay placement problem's objective function with six relays.

IV. DIFFERENTIAL EVOLUTION FOR RELAY PLACEMENT WITH A SINGLE GATEWAY

In this section, we discuss Differential Evolution (DE) which is another biology-inspired heuristic. We then show how it can be adapted to solve the relay placement problem in Section III. DE is a stochastic, parallel, direct search global optimisation method. It is robust and is often fast. DE tries to mimic the Darwinian theory of evolution based on the notion of the "survival of the fittest".

A. DIFFERENTIAL EVOLUTION (DE)

DE is an iterative algorithm like ACO. The inputs to the algorithm are similar to the inputs for the ACO algorithm: an objective function and variables over which the objective function is minimised, the variables' ranges, a procedure to evaluate the objective function for the given variable values. DE has a few other algorithm parameters which are different from ACO parameters, for example, the number of solutions generated in each iteration (also known as population), a mutation coefficient, a crossover probability, the maximum number of iterations allowed, etc. As we did for the ACO algorithm, we fix these algorithm parameters, and so do not discuss their effects on the output of the algorithm.

The DE algorithm works as follows. In the very first iteration, it generates a population of solutions for the given problem as specified by the population parameter. It then evaluates a "fitness" for each solution based on the objective function value. In the next iteration, it evolves the current population of solutions via crossovers among themselves, and generates a new population of solutions. It then compares the fitness of the new population against the fitness of the old population. Any new individual with better fitness replaces an old individual. All the other new individuals, whose fitness are worse than those of the old individuals, are dropped from the list; see the pseudo-code provided in Algorithm 1 for details. Thus, a few 'individuals' in the older population get replaced with new 'individuals' having better fitness, thereby making the solution quality of the new population better than that of the old population. This process is then repeated for a certain number of generations while constantly monitoring the solution quality across the population. The algorithm terminates early if all the individuals converge to a common solution.

As is evident from the description above, there is no gradient computation for finding the next iterate. DE can thus handle objective functions that are not necessarily continuous or differentiable. Let us now see how to use DE in the homogeneous propagation environment first before turning to the heterogeneous propagation environment.

B. DE FOR THE EUCLIDEAN STEINER TREE PROBLEM

Returning to the Euclidean Steiner Tree problem, we first generate a population for just one ($k = 1$) relay location (d_{ix}, d_{iy}). We then generate a minimum spanning tree (MST) on nodes $(d_{ix}, d_{iy}) \cup \{x_1, x_2, \ldots, x_n\}$. The link cost is taken as the Euclidean distance between two points.

Next, we calculate the fitness of each candidate relay location (d_{ix}, d_{iy}) in the population. As before, we take the fitness of the location to be the maximum of the link cost of the generated MST. (This would be equivalent to the negative of the minimum RSSI in the generated MST). The lower the maximum link cost, the better the fitness. The objective of the DE algorithm is to maximise the fitness of the population.

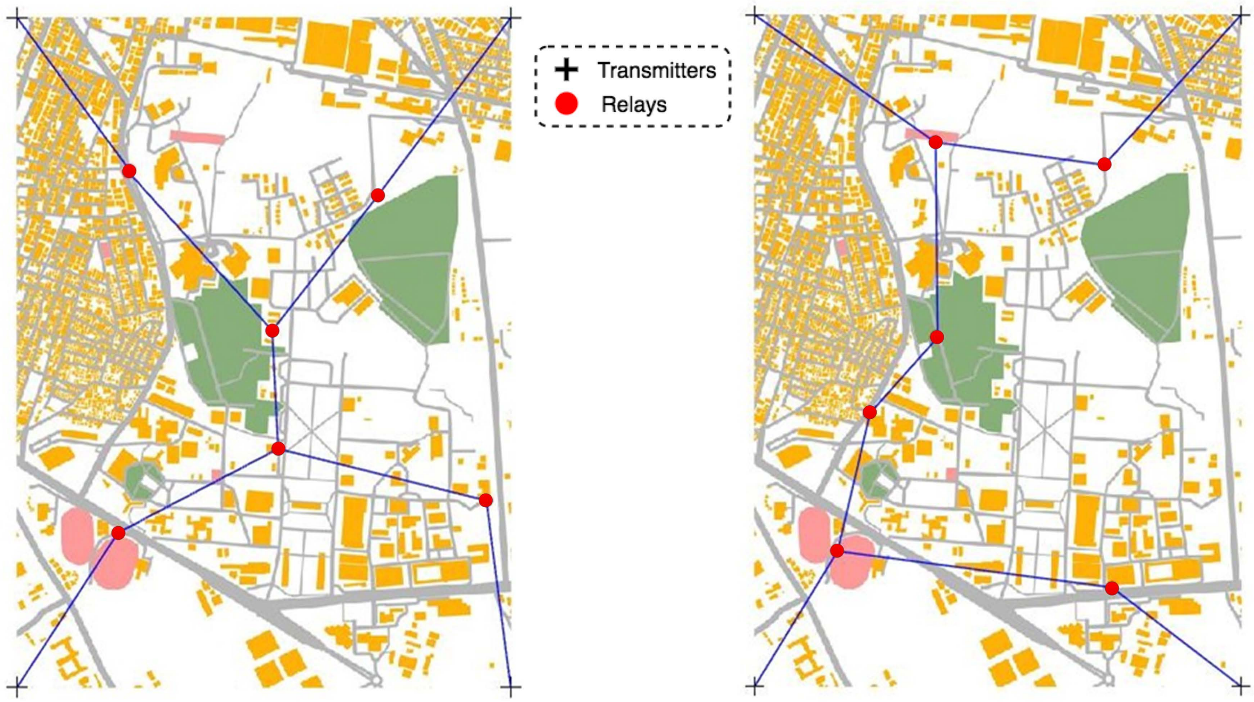

FIGURE 3. Results of the ACO algorithm for a single gateway and three transmitters in a heterogeneous region.

When the fitness stabilises across the generations, we stop the execution of the algorithm. We then increase the number of relay location by one ($k = 2$) and rerun the DE algorithm. We compare the result of $k = 2$ with $k = 1$. If the solution improves for $k = 2$ then we continue the algorithm with $k = 3$. This process continues till $k = n - 2$, the upper limit of the number of extra nodes needed for solving the Euclidean Steiner tree problem. If we do not get better results by increasing the number of relays, then we retain the solution with the previous value of k.

Reassuringly, all the solutions shown in Fig. 1 were also achieved using DE.

C. DE FOR SOLVING THE RELAY PLACEMENT PROBLEM IN A HETEROGENEOUS REGION

We now turn to our problem of relay placement in heterogeneous propagation environments. We adapt the aforementioned algorithm into one for a heterogeneous propagation environment in exactly the same way as described in Section III-C for the ACO solution framework. Again, we set the RSSI between a pair of transmitter and receiver to be the RSSI estimate put out by the algorithm in [3], which we view as an oracle. Similar to the ACO algorithm, we set the value of $\alpha = 0$ while applying the DE algorithm. While explaining DE, we did not mention the exact steps to generate a new population from the old population. There are many ways to do this, namely, rand/1/bin, rand/2/bin, best/1/bin, best/2/bin, rand-best/1/bin, etc. These different approaches are suited to different problems. For our relay placement problem, we used the rand/1/bin scheme, which is explained in more detail in the pseudo-code in Algorithm 1.

To stress-test our algorithm, as before, we placed the transmitters and a single gateway at the corners of the map. We then let DE find the best relay locations. Fig. 4 shows the results of DE when we ran it independently. The two solutions are quite different yet providing similar quality network outcomes. Furthermore, the relay locations (red nodes) are chosen to maximise the minimum RSSI of the links of the network. DE was able to connect the transmitters with five relays.

This completes the workflow description and scenario outcomes for the connectivity of the given transmitters to a single gateway or sink. In the next section, we pose the problem with multiple gateways. We then show we can use a modified algorithm to solve the multiple gateways problem.

V. HETEROGENEITY VS HOMOGENEITY: A COMPARISON WITH A BASELINE

We now demonstrate the need for handling heterogeneity in the propagation environment while building the network. Using the field measurements that were collected by us, which were reported [1], we build a channel model assuming homogeneity, estimate parameters, and then subsequently build a network using the proposed ACO and DE algorithms, using the homogeneous propagation environment model. We then compare this resulting network with the network obtained by our proposed methods assuming heterogeneity.

FIGURE 4. Results of the DE algorithm for a single gateway point and three transmitters in a heterogeneous region.

Fig. 5 shows the results obtained from running the ACO and the DE algorithms assuming the best fit homogeneous propagation model for the entire region. As before, the transmitters are located at the four corners of both the maps. The regression was done on the data from [1]. This gives us a single path-loss model for the entire region, which we then use for estimating the link quality or RSSI. In particular, only distances matter. The ACO algorithm suggests placing six relays to establish the connection between the transmitters situated at the corners of the region. The DE algorithm suggests placing seven relays to achieve connectivity. When compared with the heterogeneous estimation, the number of relays required by the ACO algorithm remains the same. But, the DE algorithm with homogeneity suggests two additional relays. Going a little further, Tables 1 and 2 highlight the difference between the RSSI estimation of the 'Homogeneous' and 'Heterogeneous' models. As evident from the tables, the differences are substantial. The estimations from the 'Homogeneous' model, second columns of Tables 1 and 2, are over-optimistic because of which both the algorithms, ACO and DE, suggest relay locations that do not work in the actual heterogeneous environments, third columns of Tables 1 and 2. The link quality when heterogeneity is taken into account is way below the RSSI threshold of -110 dBm, except in link 6 in both cases. This simulation experiment clearly shows the importance of taking heterogeneity into account while building the network.

VI. RELAY PLACEMENT PROBLEM WITH MULTIPLE GATEWAYS IN A HETEROGENEOUS REGION

Often large-scale networks that cover large deployment areas require multiple gateways. They enable connectivity over larger areas and help to distribute traffic across multiple gateways. Despite the presence of multiple gateways, transmitter nodes in large deployment areas (say city-wide deployment) may still need relays to connect to the nearest gateway among the multiple gateways available. In this section, we explain a heuristic to solve the connectivity problem when there are multiple gateways. We begin with a formal problem description.

Problem: Given a heterogeneous region, an RSSI threshold of R, n number of transmitter locations $\{x_1, x_2, \ldots, x_n\}$, m number of aggregating locations $\{y_1, y_2, \ldots, y_m\}$ in the heterogeneous region, find the minimum number of relays k and their locations $\{d_1, d_2, \ldots, d_k\}$ so that each link e of the resulting constrained Minimum Weight Forest on $S = \{y_1, y_2, \ldots, y_m\} \cup \{x_1, x_2, \ldots, x_n\} \cup \{d_1, d_2, \ldots, d_k\}$ has RSSI $\geq R$. Further, the forest must cover all the transmitter locations, every component of the forest must contain at least one of the aggregators, and the RSSI condition can be written as:

$$\min_{\substack{(i,j)\in E(cMWF(S))\\ i\neq j}} RSSI(i,j) \geq R. \tag{2}$$

Here $cMWF(S)$ is a minimum weight spanning forest that meets the spanning constraint and the constraint that every component of the forest contains at least one aggregator.

Remarks similar to those following equation (1) apply here as well. One could have searched over all possible forests meeting the minimum RSSI condition on each link. Since the max-min computation (maximum over forests, minimum over RSSIs on links, or the best forest with the highest minimum RSSI across links) is a search over a superexponential number of possibilities, our heuristic in (2) similarly

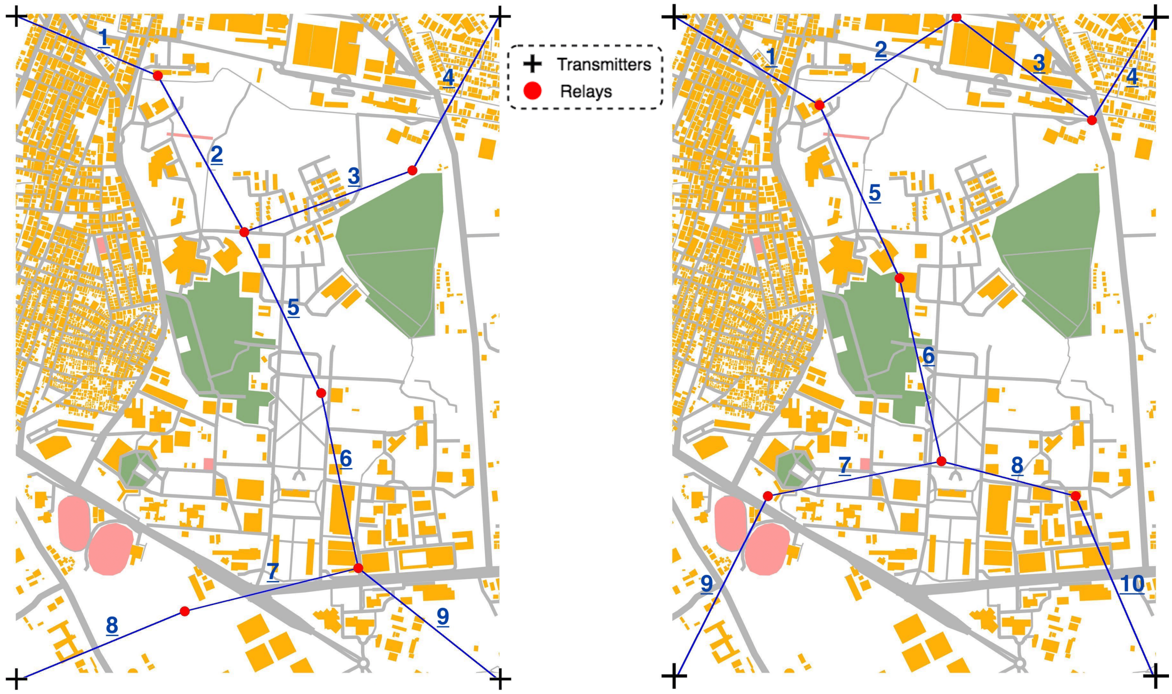

FIGURE 5. Results of the ACO algorithm (left) and the DE algorithm (right) for a single gateway point and three transmitters under the homogeneous environment assumption.

settles for the minimum weight forest meeting the minimum RSSI constraint, after applying the f transformation with a suitable α.

In the example deployments in Fig. 6 and Fig. 8, When we used the same minimum required R as in the single gateway setting (-110 dBm), we found that there was no need for relays since direct links to gateways sufficed. In order to illustrate our approach, we therefore raised the threshold to $R = -100$ dBm. The example outcomes in Fig. 6 and Fig. 8 are for this more stringent RSSI requirement.

A. SOLUTION USING ACO FOR MULTIPLE GATEWAYS: GREEDY APPROACH

Our heuristic is to follow a divide-and-conquer approach. To ensure that the constraint is satisfied, we subdivide the above-mentioned problem into smaller problems each of which is similar to the problem solved in section III-C. We then solve the smaller problems individually. The steps are as follows. First, calculate the RSSI between each transmitter and gateway. Associate each transmitter to the gateway with which it has the largest RSSI even if it is <-100 dBm. Create a cluster for each gateway by collecting all the transmitters associated with it. By doing this, we divide the entire network into smaller sub-networks corresponding to each cluster having precisely one gateway. Moreover, the use of the computed RSSI based on the propagation environment ensures that the clustering takes both heterogeneity and geographical locations into account. We can now use the same algorithm described in section III-C to find relay locations for each cluster. Fig. 6 shows the solution obtained using this approach.

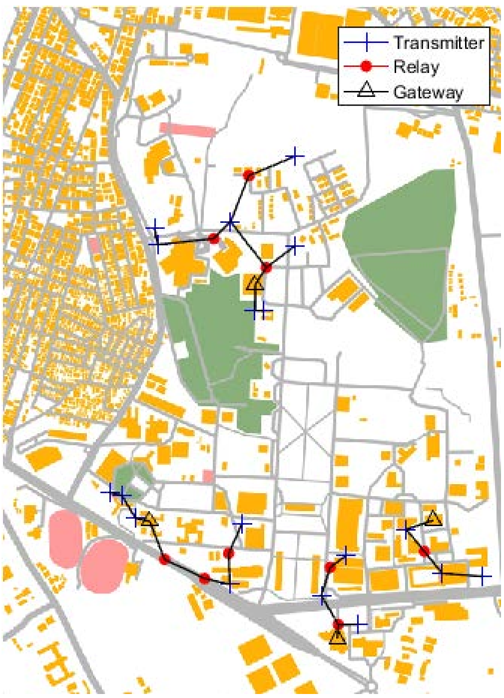

FIGURE 6. Network design for four gateways and multiple transmitters in a heterogeneous region.

In Fig. 6, all the points marked with (+) were actual locations of an in-house IoT deployment for an intelligent water distribution application that connected ground

Algorithm 1: Differential Evolution (DE) Algorithm

$k \leftarrow 0$; $n_{itr} \leftarrow$ number of iteration;
$p \leftarrow$ population; $m \leftarrow$ mutation coefficient;
$c_p \leftarrow$ crossover probability, $0 \leq c_p \leq 1$;
$V_{best} \leftarrow 0$, minimum function value;
$I_{best} \leftarrow 0$, index of minimum function value;
$p_{best} \leftarrow \varnothing$, best relay locations;

Repeat
  for $i \leftarrow 1$ **to** p **do**
    $D_i^0 = Random\{d_{i,1}^0, d_{i,2}^0, \ldots, d_{i,k}^0\}$;
    $S = \{x_1, x_2, \ldots, x_n\} \cup D_i^0$;
    $F_i^0 = \min_{\substack{(p_1,p_2) \in E(MST(S)) \\ p_1 \neq p_2}} RSSI(p_1, p_2)$;
    #[F is negative of fitness];
  end
  $(V_{best}, I_{best}) = \min F_i^0$; $p_{best} = D_{I_{best}}^0$;
  $D^0 = (D_1^0, \ldots, D_p^0)$;
  for $j \leftarrow 1$ **to** n_{itr} **do**
    for $i \leftarrow 1$ **to** p **do**
      $H_{temp} = D^{j-1} \setminus \{D_i^{j-1}\}$;
      $a, b, c = rand \in H_{temp}$;
      $d_{temp} = a + m * (b - c)$;
      #[Note: $d_{temp} = (d_{temp}(1), \ldots, d_{temp}(k))$;]
      $D_i^j \leftarrow \varnothing$;
      for $l \leftarrow 1$ **to** k **do**
        $b_{temp} = Random(0, 1)$;
        if $b_{temp} > c_p$ **then**
          $D_i^j \leftarrow D_i^j \cup d_{temp}(l)$ #[replace];
        else
          $D_i^j \leftarrow D_i^j \cup D_i^{j-1}(l)$ #[retain];
        end
      end
      $S_t = \{x_1, x_2, \ldots, x_n\} \cup D_i^j$;
      $F_{temp} = \min_{\substack{(p_1,p_2) \in E(MST(S_t)) \\ p_1 \neq p_2}} RSSI(p_1, p_2)$;
      if $F_{temp} < F_i^{j-1}$ **then**
        $D_i^j = D_i^j$; $F_i^j = F_{temp}$;
      else
        $D_i^j = D_i^{j-1}$; $F_i^j = F_i^{j-1}$;
      end
    end
    $(V_{best}, I_{best}) = \min F_i^j$; $p_{best} = D_{I_{best}}^j$;
  end
  if $V_{best} \geq R$ **then**
    print "Solution Reached";
    Output $\leftarrow p_{best}$;
    break;
  else
    $k \leftarrow k + 1$;
  end
end

Data: A set of n transmitters $T = \{x_1, x_2, \ldots, x_n\}$ where each x_i, $1 \leq i \leq n$ represents a transmitter location in the deployment region and RSSI Threshold R.
Result: A set of k relays $D = \{d_1, d_2, \ldots, d_k\}$ where each d_i, $1 \leq i \leq k$ represents a relay location in the deployment region.

TABLE 1. Link quality estimation difference between heterogeneous and homogeneous environment: ACO algorithm.

Link No	RSSI Homogeneous (dBm)	RSSI Heterogeneous (dBm)
1	-105.8014	-121.6824
2	-108.3341	-117.6230
3	-108.3521	-118.7415
4	-108.0741	-134.6670
5	-108.2856	-124.2164
6	-108.3412	-99.4867
7	-108.3440	-137.8975
8	-108.3522	-146.0792
9	-108.3478	-134.4170

TABLE 2. Link quality estimation difference between heterogeneous and homogeneous environment: DE algorithm.

Link No.	RSSI Homogeneous (dBm)	RSSI Heterogeneous (dBm)
1	-107.4510	-122.6603
2	-106.9426	-114.4766
3	-107.5933	-126.8801
4	-102.5357	-132.6543
5	-109.2834	-109.5772
6	-109.0543	-110.2381
7	-108.1727	-113.6653
8	-104.5568	-127.1674
9	-110.3977	-152.1958
10	-110.0066	–115.9396

level reservoirs, overhead tanks, and flow meters [41]. Gateway locations are marked with the triangle symbol. The relays are identified by the ACO algorithm by first dividing the multiple-gateway relay-placement problem into four single-gateway subproblems, and then each of the smaller problems using the previously outlined solutions. Our multiple-gateway algorithm suggested the use of nine relays. Their locations are marked with the red circles in Fig. 6. Observe that we have much more efficient and localised networks. Fig. 7 zooms into the cluster on the north side of the IISc Campus. This subproblem has solution with three relays which are shown by the red dots in the figure (nodes 9, 10, 11). As Table 3 shows, all links are above the minimum RSSI threshold R of -100 dBm. Note that one should not compare this solution with the solutions of Fig. 3 or 4 which are for a different set of transmitter and gateway

Algorithm 2: Algorithm for Generating Subnetworks Based on RSSI Matrix

Initialisation;
$S_{Tx} \leftarrow \{x_1, x_2, \dots, x_n\}$;
$S_{Snk} \leftarrow \{y_1, y_2, \dots, y_m\}$;

$D_{Tx,Snk} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** n **do**
  $D' \leftarrow \emptyset$;
  for $j \leftarrow 1$ **to** m **do**
    $D' = D' \cup RSSI\{x_i, y_j\}$;
  end
  $D_{Tx,Snk} = D_{Tx,Snk} \cup D'$;
end
$\hat{G}_{Subnetworks} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** m **do**
  $G'_{Subnetwork} \leftarrow \emptyset$;
  for $j \leftarrow 1$ **to** n **do**
    if $\min D_{Tx=j,Snk=\forall i} = D_{Tx=j,Snk=i}$ **then**
      $G'_{Subnetwork} = G'_{Subnetwork} \cup j$;
  end
  $\hat{G}_{Subnetworks} = \hat{G}_{Subnetworks} \cup G'_{Subnetwork}$;
end
Output: $\hat{G}_{Subnetworks}$;

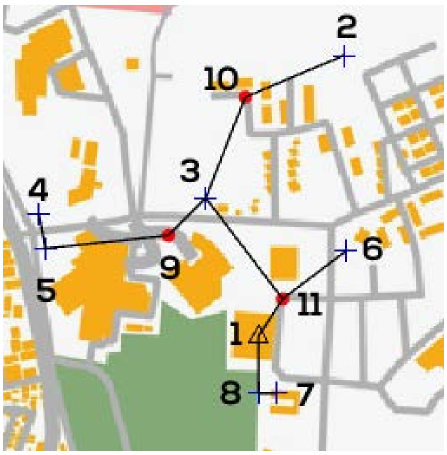

FIGURE 7. The network design for North side of the IISc campus using ACO.

locations, and more importantly, for a lower RSSI requirement of $R = -110$ dBm.

B. SOLUTION USING DE FOR MULTIPLE GATEWAYS: GREEDY APPROACH

We now apply the same divide-and-conquer approach of Section VI-A, and solve the subproblems using the DE algorithm. Fig. 8 shows the simulation outcomes. The number of relays needed using both ACO and DE are nine, but the identified locations are different. Fig. 9 zooms into the cluster on the North-side of the IISc Campus. This subproblem also

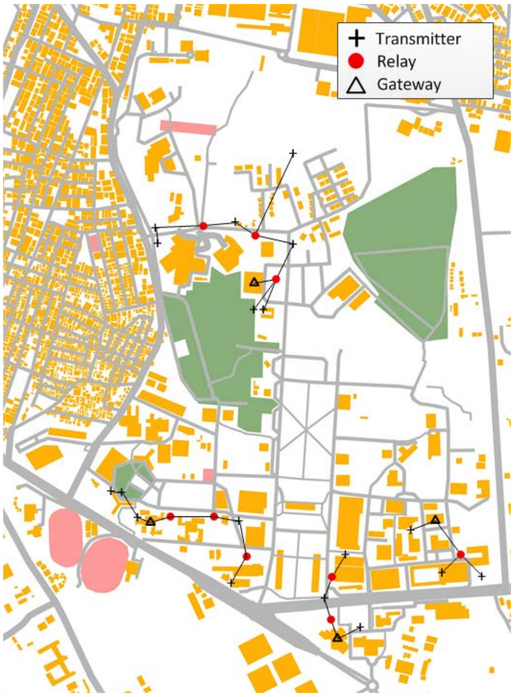

FIGURE 8. Network design for four gateways and multiple transmitters in a heterogeneous region, using the DE algorithm.

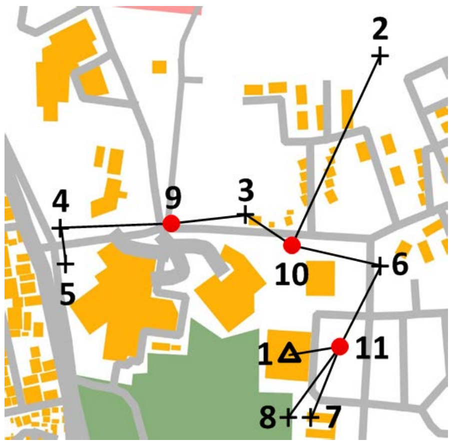

FIGURE 9. The network design for the North side of the IISc campus, using the DE algorithm.

requires three relay nodes for local connectivity. Table 4 shows the observed RSSI between each transmitter-receiver pair. Notice that all the links in both deployments have RSSI greater than RSSI threshold R of -100 dBm. The link RSSIs are however a little more balanced than in Table 4 suggesting that, in this case, the DE solution may be preferable to the ACO solution.

C. DISCUSSION

The algorithms described in subsections VI-A and VI-B start with a certain way of dividing the larger problem into smaller problems, and might lead to sub-optimal solutions. One hypothetical situation is shown in Fig. 10 where our earlier

TABLE 3. Network connectivity for ACO.

Node Id (Tx)	Node Id (Rx)	RSSI Heterogeneous (dBm)
1	8	-99.1163
1	11	-95.7368
2	10	-85.7739
3	9	-74.9758
3	10	-85.8552
3	11	-87.83436
4	5	-69.9692
5	9	-88.9698
6	11	-80.6244
7	8	-77.2838

TABLE 4. Network connectivity for DE.

Node Id (Tx)	RSSI Node Id (Rx)	RSSI Heterogeneous (dBm)
1	11	-85.80281
2	10	-85.3519
3	9	-87.5490
4	9	-81.2334
5	4	-71.8457
8	11	-88.5876
10	3	-89.2899
10	6	-87.2158
11	6	-88.8446
11	7	-86.2015

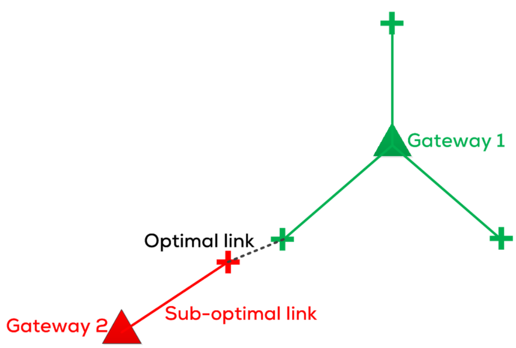

FIGURE 10. A hypothetical scenario leading to a sub-optimal solution when the divide-and-conquer approach is used.

proposed divide-and-conquer approach would associate the red transmitter (+) to the red gateway 2 and the green transmitters to the green gateway 1. However, a better solution in this situation is to connect all the transmitters (+) to the green gateway 1 and to let the red gateway 2 operate by itself. Our proposed divide-and-conquer procedure is a local approach to identify the smaller single-gateway sub-problems whereas the objective in equation (2) requires global considerations. In the next subsections, we discuss two remedies.

D. MODIFICATION BY RESHUFFLING

In this section, we propose an improvement over the basic divide-and-conquer technique of Sections VI-A and VI-B and address the sub-optimal scenario highlighted in Section VI-C. This algorithm introduces additional steps after the generation of individual sub-networks. Once the sub-networks are generated along with the relay locations, the algorithm does the following for each non-sink node: identify the nearest neighbour, $NN1$, from the current sub-network and the nearest neighbour, $NN2$, from the full network. If $NN1 = NN2$, do nothing and move to the next node. If $NN1 \neq NN2$, assign the current node to the sub-network containing $NN2$. We call this step a 'reshuffling' step. One thing to note here is that we never reassign the sinks (gateways) using this reshuffling step. Reassigning a sink to another network results in one sub-network with two sinks and another with no sink. Every reshuffling results in a change to two subnetworks (a subnetwork that loses a node and a subnetwork that gains a node). The single-sink algorithm is then re-run on the two subnetworks before proceeding with the next node. The algorithm stops when there are no nodes to be shuffled or when we reach a repeated configuration.

E. MODIFICATION BY ELIMINATION FROM MINIMUM SPANNING TREE

In this section, we propose an alternative algorithm to partition the network into a collection of sub-networks. In the first step, the algorithm generates a minimum spanning tree consisting of all the transmitters and all the sinks. To identify the sub-networks, we remove certain links as per the following steps. First, for each pair of sinks, we find the path that connects them. From the union of these paths, we pick the link with the lowest RSSI and remove it. As a consequence the number of sub-networks increases by one, and one pair of sinks is separated. We then repeat the procedure in each sub-network that has two or more sinks. This procedure is guaranteed to stop after a removal of $N-1$ links where N is the number of sinks. Further, this procedure will end with precisely one sink per sub-network. The single sink solution methodology is then applied on each of the sub-networks.

F. A COMPARISON OF THE MODIFICATIONS

In this subsection, we explain the results obtained by employing the algorithms explained in Section VI-D and VI-E. The results can be seen through the lens of many metrics such as the RSSI of the weakest link in all the sub-networks, the average number of hops to reach a sink, the average number of transmitter nodes per sink, etc. The outcomes of the algorithms are shown in Fig. 11 to 16.

In the first set of figures, Fig. 11 and 14, the result obtained from both the algorithms are the same. Even though both the algorithms started with a different initial state, they converged to the same solution.

In the next set of figures, Fig. 12 and 15, the results obtained from both the algorithms are quite different. Here, the sub-networks obtained using the 'reshuffling' algorithm are well segregated (better distribution of transmitter nodes to sinks) and with a lower average number of hops to reach

Algorithm 3: Reshuffling Based Approach

Initialisation;
$S_{Tx} \leftarrow \{x_1, x_2, \ldots, x_n\}$;
$S_{Snk} \leftarrow \{y_1, y_2, \ldots, y_m\}$;

$D_{Tx,Snk} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** n **do**
  $D' \leftarrow \emptyset$;
  for $j \leftarrow 1$ **to** m **do**
    $D' = D' \cup RSSI\{x_i, y_j\}$;
  end
  $D_{Tx,Snk} = D_{Tx,Snk} \cup D'$;
end
$\hat{G}_{Subnetworks} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** m **do**
  $G'_{Subnetwork} \leftarrow \emptyset$;
  for $j \leftarrow 1$ **to** n **do**
    if $\min D_{Tx=j,Snk=\forall i} = D_{Tx=j,Snk=i}$ **then**
      $G'_{Subnetwork} = G'_{Subnetwork} \cup j$;
  end
  $\hat{G}_{Subnetworks} = \hat{G}_{Subnetworks} \cup G'_{Subnetwork}$;
end
$S_{all} \leftarrow S_{Tx} \cup S_{Snk}$;
$G \leftarrow CompleteGraph(S_{all})$;
Flag = True;
while *Flag* **do**
  Flag = False;
  for $i \leftarrow 1$ **to** n **do**
    $nn = NearestNeighbour(x_i \in G)$;
    for $j \leftarrow 1$ **to** $|\hat{G}_{Subnetworks}|$ **do**
      if $i \in \hat{G}_{Subnetworks}(j)$ **then**
        $subnetwork_1 = i$;
      if $nn \in \hat{G}_{Subnetworks}(j)$ **then**
        $subnetwork_2 = nn$;
    end
    if $subnetwork_1 \neq subnetwork_2$ **then**
      $\hat{G}_{Subnetworks}(subnetwork_1) = \hat{G}_{Subnetworks}(subnetwork_1) \notin \{i\}$;
      $\hat{G}_{Subnetworks}(subnetwork_2) = \hat{G}_{Subnetworks}(subnetwork_2) \cup \{i\}$;
      Flag = True;
      break;
  end
end
Output: $\hat{G}_{Subnetworks}$;

---

the sink, for e.g., node 3 is 6 hops away from the designated sink node 21 in Fig. 12 while the same node is 8 hops away from the designated sink node 23 in Fig. 15. Thus, the sub-networks obtained with MST based approach leads to a solution where one sink is connected to the majority of the transmitter nodes, and other sinks are connected to three transmitter nodes at best. This solution also leads to a higher number of hops to reach the sink. So, for this distribution of transmitters and sinks, the 'reshuffling' algorithm leads to better sub-networks.

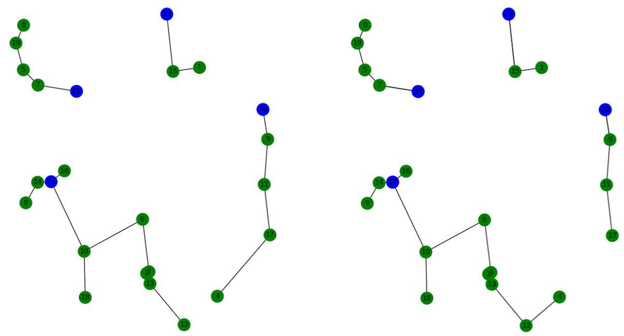

FIGURE 11. Reshuffling using nearest neighbour (NN) for the multi-gateway connectivity.

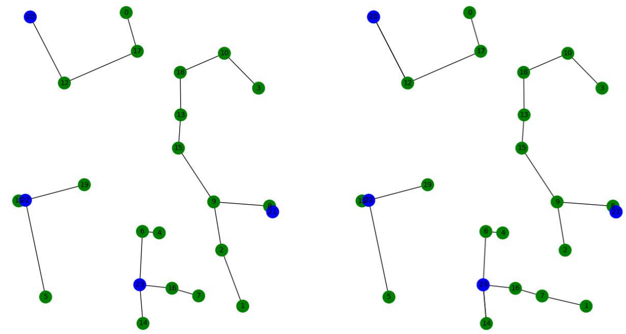

FIGURE 12. NN Based Approach: The initial configuration (left) and the final network configuration (right).

In the last set of figures, Fig. 13 and 16, the situation is reversed, and the MST-based approach fares better than the 'reshuffling' algorithm. Notice that in Fig. 16, node 18 is 3 hops away from its assigned sink node 20 whereas the same node is 10 hops away from its assigned sink node 22 in Fig. 13. Here, the resulting sub-networks arising from the MST-based approach has a better distribution of transmitter nodes per sink, with lower average number of hops to reach the sink.

These sets of results show that both algorithms result in better sets of sub-networks for different distributions of transmitter nodes and sinks. The end-user has to define other criteria/metric to evaluate the resulting sub-networks and choose the one which fares better on that metric.

VII. DIFFERENTIAL EVOLUTION FOR COVERAGE OPTIMISATION

In coverage problems, the network is designed not for connecting a fixed or static set of transmitters but to accommodate future transmitters that may arise at any location in the deployment region. We can then have a layer of separation between IoT application developers (who develop innovative applications like water distribution monitoring,

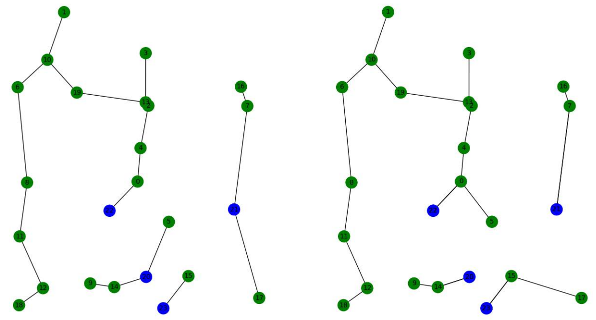

FIGURE 13. NN Approach with the initial configuration on the left hand side and the final configuration on the right hand side.

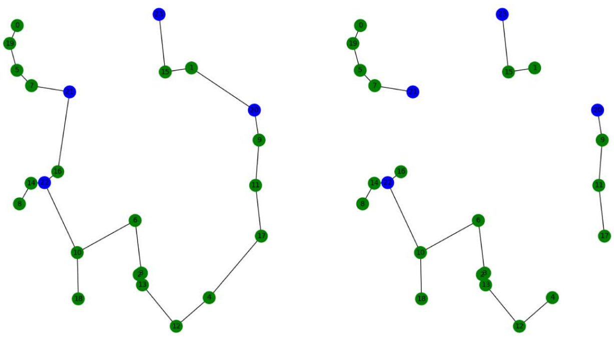

FIGURE 14. The MST approach for the multi-gateway connectivity.

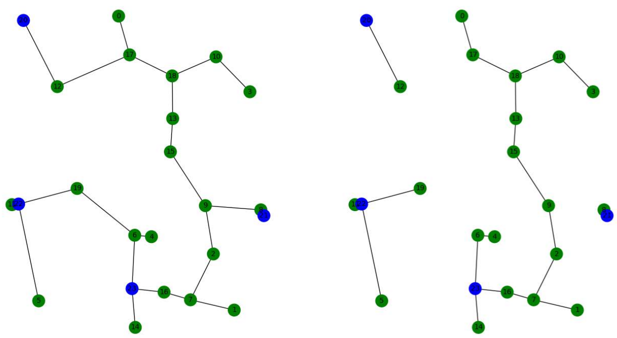

FIGURE 15. MST approach results in a worse configuration than NN approach outcome.

energy systems monitoring, etc.) and IoT data transport service providers (who focus on the coverage problem, e.g., LoRaWAN [42]). IoT application developers then have the flexibility to evolve their end-application service offerings to adapt to changing requirements, knowing that the IoT data transport service provider has deployed a network with good coverage. In this section, we try to address the coverage problem in a heterogeneous propagation environment. The formal description of the problem is as follows. Relays now become base stations.

Problem: Given a heterogeneous deployment region, an RSSI threshold R, and a minimum coverage percentage of $P\%$, find the minimum number of base stations required to provide the $P\%$ coverage of the heterogeneous region with RSSI $\geq R$ on the covered region.

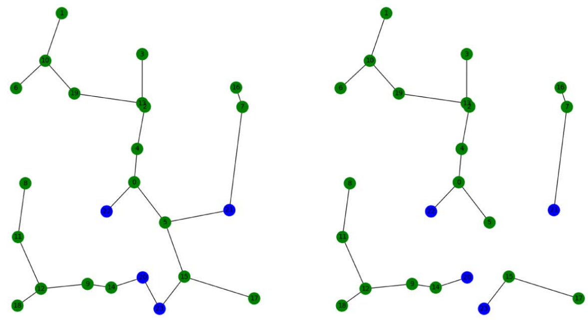

FIGURE 16. MST approach from the initial configuration on the left hand side to the final outcome on the right hand side.

To tackle this problem, we fall back to Algorithm 1 but with a modified cost function. The cost function used to solve this problem is the uncovered fraction and is explained in Algorithm 5. The function $Coverage(x_i)$ is the region covered by a base station located at x_i and takes into account the heterogeneity of the deployment region. The DE algorithm attempts to minimise this cost and thus maximise coverage.

The modified objective introduces computational challenges at two levels, which we now describe.

First, given N base stations, we need to compute the coverage fraction, which requires the computation of the coverage area of each base station. This has to be done in a brute-force way because of heterogeneity. We, however, note that the heterogeneity does not render the RSSI to be completely unstructured. This is because the deployment region can be usually partitioned into reasonably regular component subsets with homogeneous propagation parameters within each subset. We, therefore, divide the deployment region into smaller cells of a fixed shape capable of tiling the entire region; we used a square. Each cell is sufficiently small that we may assume uniform propagation conditions within the cell. Each cell also has a representative point (e.g., the centre of the cell in case of a square). The RSSI between a pair of points is taken to be the RSSI between the centres of cells containing the points under consideration. Additionally, we proceed in a spiral fashion around the candidate base station until we are able to completely enclose the base station with cells of RSSI lower than the target R, and declare the coverage region as the subset of this region with RSSI $\geq R$. We use the oracle in [3] to estimate the RSSI. This simplifies the computation of the coverage region of a base station to some extent, but it is nevertheless computationally intensive, O(number of cells).

The second computational difficulty is at the level of the differential evolution algorithm. Being an iterative algorithm, DE introduces additional computation overhead with a multiplication factor of $p \cdot n_{itr}$, where p is the population size and n_{itr} is the number of iterations. This means that during a single run of the algorithm, during which the number of base stations N is gradually increased from 1 to k, we will

Algorithm 4: Minimum Spanning Tree Based Approach

Initialisation;
$S_{Tx} \leftarrow \{x_1, x_2, \ldots, x_n\}$;
$S_{Snk} \leftarrow \{y_1, y_2, \ldots, y_m\}$;

Derived quantities;
$S_{all} \leftarrow S_{Tx} \cup S_{Snk}$;
$MST \leftarrow Minimum\ Spanning\ Tree(S_{all})$;
$S_{SnkPairs} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow (i+1)$ **to** m **do**
    $S_{SnkPairs} = S_{SnkPairs} \cup \{y_i, y_j\}$;
  end
end
#[Note: $|S_{SnkPairs}| = (m(m-1)/2)$];
$S_{paths} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** $|S_{SnkPairs}|$ **do**
  $T_{x1} = S^i_{SnkPairs}(1)$;
  $T_{x2} = S^i_{SnkPairs}(2)$;
  $E_{T_{x1},T_{x2}} = Path_{T_{x1},T_{x2}}(MST)$;
  $S_{paths} = S_{paths} \cup E_{T_{x1},T_{x2}}$;
end
$num_{subnetworks} \leftarrow 1$;
$G_{Subnetworks} = S_{paths}$;
while $num_{subnetworks} \neq m$ **do**
  for $\forall H_{subnetwork} \in G_{Subnetworks}$ **do**
    $N_{Snk} \leftarrow 0$;
    for $V \in H_{subnetwork}$ **do**
      if $V \in S_{Snk}$ **then**
        $N_{Snk} \leftarrow N_{Snk} + 1$;
    end
    if $N_{Snk} > 1$ **then**
      $E_{weak} \leftarrow 0$;
      for $E \in H_{subnetwork}$ **do**
        if $E < E_{weak}$ **then**
          $E_{weak} \leftarrow E$;
      end
      $G_{subnetwork} \leftarrow G_{subnetwork} \setminus E_{weak}$;
      $num_{subnetworks} \leftarrow num_{subnetworks} + 1$;
  end
end
Output: $G_{Subnetworks}$;

compute the coverage area for a total of $p \cdot n_{itr} \cdot \sum_{N=1}^{k} N = p \cdot n_{itr} \cdot (k \cdot (k+1)/2)$ number of base station locations, where k is the minimum number of base stations required to cover $P\%$ of given heterogeneous region with $RSSI \geq R$.

To speed up the running time of our algorithm, we optimised the combination of the two steps as follows: we generated the grid of possible transmitter locations, computed the coverage area of the transmitters at each of these grid locations, and saved the coverage area matrix with RSSI values for each cell of the grid (termed pixel). Subsequently, while running the DE algorithm, we directly fetched the coverage

Algorithm 5: Cost Function for the Coverage Problem. The Function $Coverage(x_i)$ Is the Region Covered by a Base Station at x_i

Trial Solution;
$S_{Trial} \leftarrow \{x_1, x_2, \ldots, x_n\}$;

$S_{Cover} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** n **do**
  $S' = Coverage(x_i)$;
  $S_{Cover} = S_{Cover} \cup S'$;
end
Output: $Cost = 1 - S_{Cover}/Total\ Area$;

matrix entries associated with these possible transmitter locations. These steps helped us avoid redundant coverage area calculations and thus improved the running time of our algorithm to some extent.

A. RESULTS

The results obtained using the above described coverage algorithm are shown below in Fig. 18–20. The results are generated with the following parameters: transmission power of 14 dBm, RSSI threshold R of -110 dBm, and a target coverage percentage $P = 90\%$. The entire deployment area was subdivided into 620×1080 pixels, and each cell of the partition comprised 20×20 pixels.

Fig. 18 shows two distinct sets of results obtained for providing coverage in the heterogeneous regions of the IISc Campus, shown in Fig. 17, which is spread over 170 hectares (each pixel is of area roughly $2.5\,m^2$ and so each cell is of area $1000\,m^2$). Both figures demonstrate achievement of the targeted coverage (90%) for the given minimum RSSI requirement with three base stations. In both the results, the algorithm chose to maximise the coverage area by prioritising the coverage of a moderately wooded area, which is the largest portion of the propagation environment, and avoided the highly residential buildings area towards the upper portion of the map. Before we return to discuss the uncovered area, let us explore another example, this time a town in South India.

Fig. 20 shows the results for providing coverage in Kakinada, the sixth-largest city in the Indian state of Andhra Pradesh. The Kakinada deployment has a larger and heterogeneous deployment area than the IISc Campus. We provide two candidate outcomes using the DE algorithm. Both outcomes achieve the target coverage (90%) at the specified RSSI threshold with seven base stations. One point to note is that we had only partial GIS data available for the Kakinada town. Rich GIS categorisation, such as that available for the IISc Campus on OpenStreetMaps, was not yet available for the Kakinada. There were only three available categorisations of Kakinada – buildings, open area, and roads, as shown in Fig. 19. The GIS map plays a very crucial role in predicting the coverage area and explains the surprisingly few number of transmitters required for providing coverage to 90% of

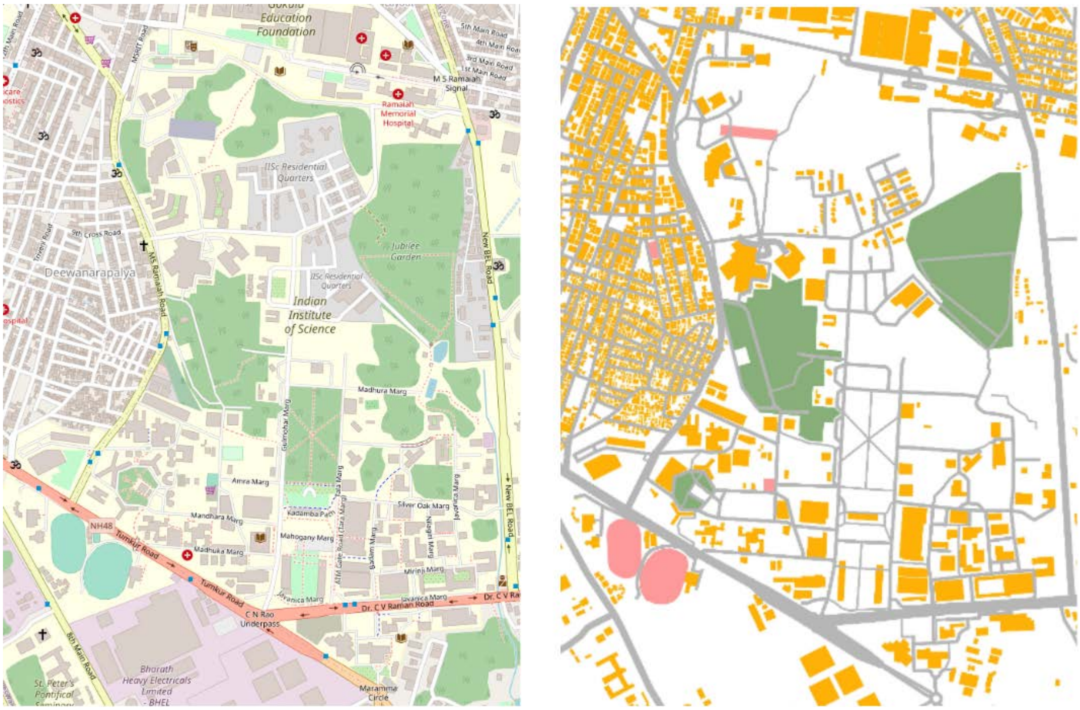

FIGURE 17. The Indian Institute of Science Campus. The picture on the right is colour-coded to show the layers of the GIS data. There are five layers—open area, moderately wooded area, heavily wooded area, roads, and buildings.

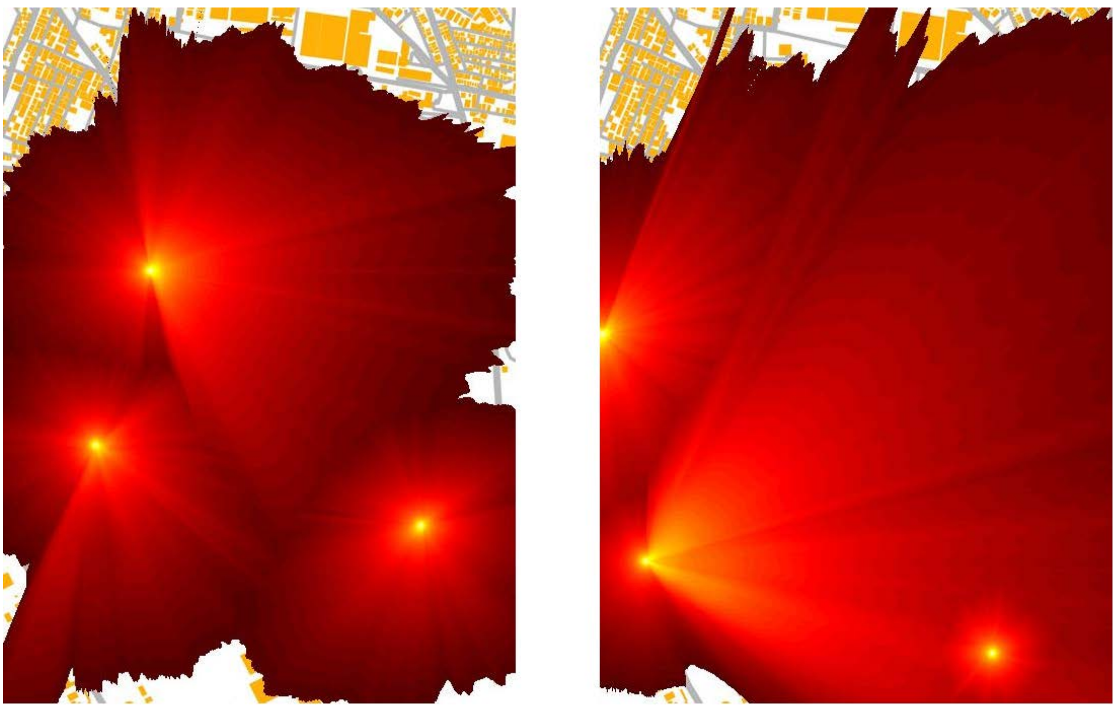

FIGURE 18. Two different solutions for coverage of the heterogeneous IISc campus.

Kakinada City. The algorithm converges to a solution where all the recommended transmitter locations are all in the open area. Again, note that the buildings areas are left uncovered by the algorithm. A re-weighted objective can increase the importance of these areas for better coverage. Comparing the two sets of results, there are a couple of transmitters placed at similar locations. This suggests that the significance of these locations in achieving good coverage of Kakinada.

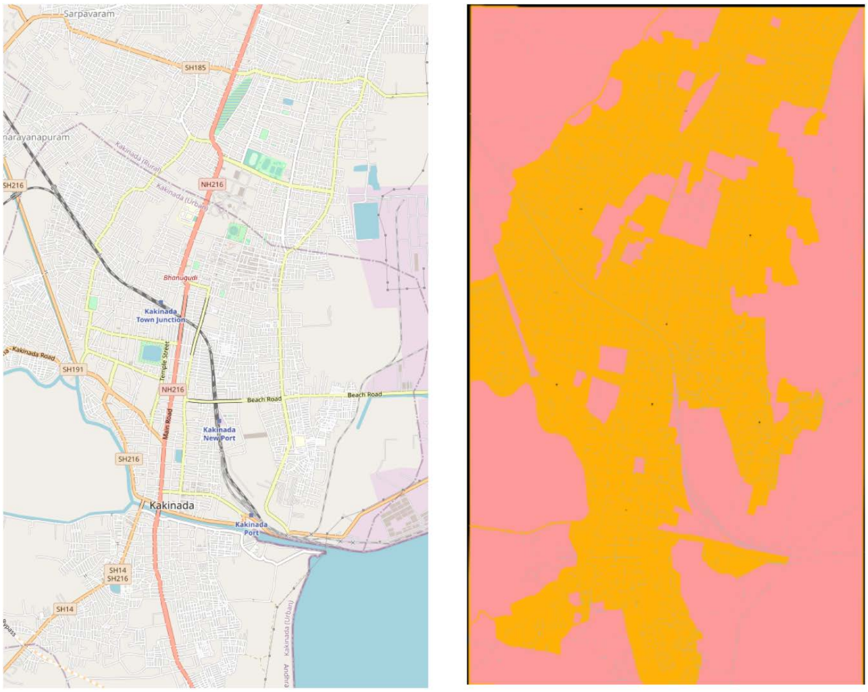

FIGURE 19. Kakinada city: the sixth largest in the Indian state of Andhra Pradesh with its GIS layered map.

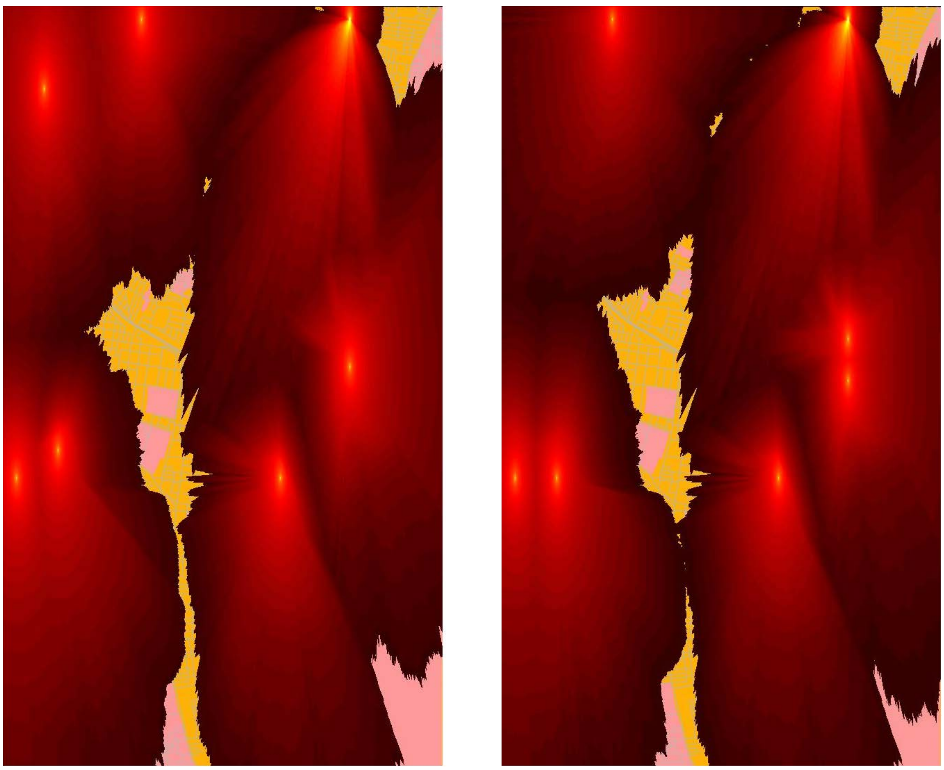

FIGURE 20. Two different solutions for the coverage of Kakinada city with seven transmitters, assuming heterogeneity.

Similar to the results of the IISc campus, the area which is not covered by the both the solutions remains largely the same between both the two solutions for the Kakinada town.

B. COMPARISON WITH THE GIBBS-SAMPLING BASED ALGORITHM

We compare our algorithm for coverage with the Gibbs-sampling based algorithm of [25]. Under Gibbs-sampling a

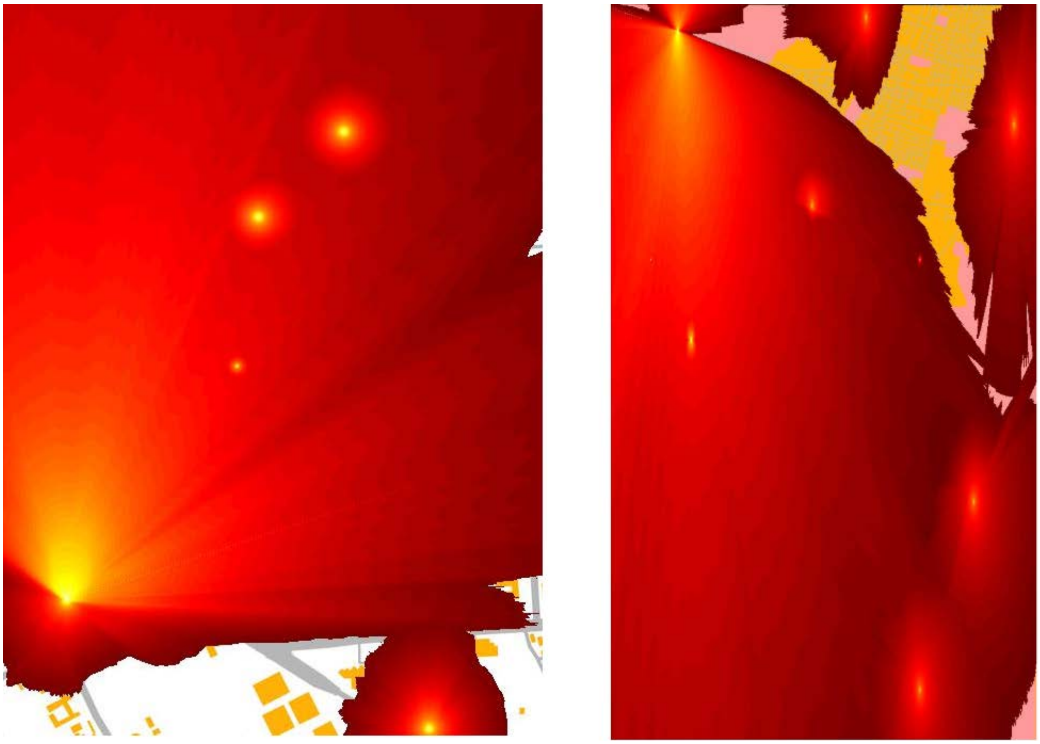

FIGURE 21. Gibbs sample-based optimisation for coverage of the IISc Campus (left) and Kakinada city (right).

random node is selected, its position is perturbed randomly and the new position is either accepted or rejected with a certain probability that depends on the improvement of the new solution. Fig. 21 shows the results of the Gibbs-sampling based algorithm for coverage of the IISc campus and the City of Kakinada. For the IISc campus, the left subfigure of Fig. 21, the Gibbs sampling-based algorithm suggests placing five transmitters while our DE algorithm is able to achieve coverage with three transmitters (for a given minimum RSSI). For coverage of Kakinada, the right subfigure of Fig. 21, the Gibbs-sampling based algorithm suggests using nine transmitters while our DE-based algorithm uses only seven.

In both the comparisons, the DE-based algorithm outperforms Gibbs-sampling based algorithm. In both the cases, required number of transmitters are more than those of the DE Algorithm. While both the algorithms are iterative in nature, the search space that the DE-based algorithm explores is larger than that of Gibbs-sampling based algorithm and therefore yield a better outcome.

If the areas which are uncovered are of importance and should be covered, one can adapt our approach by assigning weights to each pixel (in terms of its importance), modifying the cost to be the total weight of the uncovered pixels, and then using our coverage algorithm with this modified cost function. This adaptation is straightforward and is therefore not pursued here.

VIII. CONCLUSION

In this paper, we first introduced the relay placement problem in a heterogeneous region and articulated the challenges in solving it in the setting of a heterogeneous propagation environment. We solved this problem using a two-step approach. First, we used an oracle function to provide the RSSI across a pair of transmitter and receiver points which handled the complexity arising from heterogeneity. Next, we used either the Ant Colony Optimisation (ACO) or the Differential Evolution (DE) heuristic to tackle the intractability of the resulting minimum cost Steiner tree problem. We then combined the two steps to solve the relay placement problem in single gateway scenarios under heterogeneous propagation environments, and highlighted the results obtained from the ACO and the DE algorithms. We then extended the approach to the problem with multiple gateways using a divide-and-conquer approach. We highlighted a problem associated with a local clustering approach and proposed two modifications: reshuffling and minimum spanning tree pruning. We then compared the performance of these two approaches through the lens of objectives such as the average number of hops to reach the sink and the average number of transmitter nodes per sink. In some situations, the reshuffling approach works better, while in other situations, the pruning approach is better. We then showed how to deploy for connectivity in the IISc Campus.

We also studied the coverage area problem in the setting of heterogeneous propagation environments. We highlighted the computational challenges, which are partly due to the modified cost function and partly due to the heterogeneity of the propagation environment. We explained optimisations arising from pre-computation of the coverage area for the gridded transmitter locations in order to improve the running time of the algorithm. We highlighted satisfactory solutions for deployment in two different deployment regions – the IISc campus and Kakinada, Andhra Pradesh, India. We ran two instances in each case and found that the number of relays/base-stations were in agreement.

In summary, the automatic deployment workflows that we have created, for providing connectivity and coverage in heterogeneous propagation environments, can save significant time- and engineering-resources, thereby enabling widespread outdoor IoT deployments.

DATA SHARING

The computer code that generated the outcomes of this study is accessible from [43]. The data used in this study are available from [44], [45].

ACKNOWLEDGMENT

An earlier version of this paper was presented in part at the 12th IFIP Wireless and Mobile Networking Conference: Med-Com-Net Special Track, Paris, France, in September 2019 [1] [DOI: 10.23919/WMNC.2019.8881829]. The authors would like to thank Rashmi Ballamajalu from the Indian Institute of Science for help with the field experiments. The data collected from these experiments powered the RSSI prediction engine. The authors would also like to thank Renu Subramanian for providing assistance in GIS map generation and simulations.

REFERENCES

[1] N. Rathod and R. Sundaresan, "Relay placement algorithms for communication in a heterogeneous propagation environment," in *Proc. 12th IFIP Wireless Mobile Netw. Conf. (WMNC)*, Paris, France, Sep. 2019, pp. 219–226.

[2] N. Rathod, P. Jain, R. Subramanian, S. Yawalkar, M. Sunkenapally, B. Amrutur, and R. Sundaresan, "Performance analysis of wireless devices for a campus-wide IoT network," in *Proc. 13th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2015, pp. 84–89.

[3] N. Rathod, R. Subramanian, and R. Sundaresan, "Coverage estimation in outdoor heterogeneous propagation environments," *IEEE Access*, vol. 8, pp. 31660–31673, 2020.

[4] F. K. Hwang, D. S. Richards, P. Winter, and P. Widmayer, "The Steiner tree problem, annals of discrete mathematics, volume 53," *ZOR-Methods Models Oper. Res.*, vol. 41, no. 3, p. 382, 1995.

[5] A. Bhattacharya and A. Kumar, "A shortest path tree based algorithm for relay placement in a wireless sensor network and its performance analysis," *Comput. Netw.*, vol. 71, pp. 48–62, Oct. 2014.

[6] A. Bhattacharya, S. M. Ladwa, R. Srivastava, A. Mallya, A. Rao, D. G. R. Sahib, S. V. R. Anand, and A. Kumar, "SmartConnect: A system for the design and deployment of wireless sensor networks," in *Proc. 5th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2013, pp. 1–10.

[7] P. Crescenzi, V. Kann, and M. Halldórsson. (1995). *A Compendium of NP Optimization Problems*. [Online]. Available: https://www.csc.kth.se/tcs/compendium/

[8] M. Nikolov and Z. J. Haas, "Relay placement in wireless networks: Minimizing communication cost," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3587–3602, May 2016.

[9] A. J. Perez, M. A. Labrador, and P. M. Wightman, "A multiobjective approach to the relay placement problem in WSNs," in *Proc. IEEE wireless Commun. Netw. Conf.*, Mar. 2011, pp. 475–480.

[10] A. Efrat, S. P. Fekete, P. R. Gaddehosur, J. S. Mitchell, V. Polishchuk, and J. Suomela, "Improved approximation algorithms for relay placement," in *Proc. Eur. Symp. Algorithms*. Berlin, Germany: Springer, 2008, pp. 356–367.

[11] M. Minelli, M. Ma, M. Coupechoux, J.-M. Kelif, M. Sigelle, and P. Godlewski, "Optimal relay placement in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 998–1009, Feb. 2014.

[12] J.-S. Li, H.-C. Kao, and J.-D. Ke, "Voronoi-based relay placement scheme for wireless sensor networks," *IET Commun.*, vol. 3, no. 4, pp. 530–538, 2009.

[13] B. Lin, P. H. Ho, L. L. Xie, X. Shen, and J. Tapolcai, "Optimal relay station placement in broadband wireless access networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 2, pp. 259–269, Feb. 2010.

[14] F. M. Al-Turjman, H. S. Hassanein, and M. Ibnkahla, "Optimized relay placement to federate wireless sensor networks in environmental applications," in *Proc. 7th Int. Wireless Commun. Mobile Comput. Conf.*, Jul. 2011, pp. 2040–2045.

[15] K. Nitesh and P. K. Jana, "Relay node placement with assured coverage and connectivity: A Jarvis March approach," *Wireless Pers. Commun.*, vol. 98, no. 1, pp. 1361–1381, Jan. 2018.

[16] S. Sapre and S. Mini, "Optimized relay nodes positioning to achieve full connectivity in wireless sensor networks," *Wireless Pers. Commun.*, vol. 99, no. 4, pp. 1521–1540, Apr. 2018.

[17] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. Distrib. Auton. Robotic Syst.* Tokyo, Japan: Springer, 2002, pp. 299–308.

[18] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, "A mobile network planning tool based on data analytics," *Mobile Inf. Syst.*, vol. 2017, pp. 1–16, Feb. 2017.

[19] F. P. Quint ao, F. G. Nakamura, and G. R. Mateus, "Evolutionary algorithm for the dynamic coverage problem applied to wireless sensor networks design," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Sep. 2005, pp. 1589–1596.

[20] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Self-deployment algorithms for coverage problem in a network of mobile sensors with unidentical sensing ranges," in *Proc. IEEE Global Telecommun. Conf. GLOBECOM*, Dec. 2010, pp. 1–6.

[21] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Comput. Commun.*, vol. 29, no. 4, pp. 413–420, Feb. 2006.

[22] F. Tossa, W. Abdou, E. C. Ezin, and P. Gouton, "Improving coverage area in sensor deployment using genetic algorithm," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2020, pp. 398–408.

[23] S. Y. Shin and I. F. M. Zain, "Cellular network planning for heterogeneous network using geo-clustering algorithm," in *Proc. Int. Conf. ICT Converg. (ICTC)*, Oct. 2013, pp. 448–449.

[24] J. Gazda, E. Šlapak, G. Bugár, D. Horváth, T. Maksymyuk, and M. Jo, "Unsupervised learning algorithm for intelligent coverage planning and performance optimization of multitier heterogeneous network," *IEEE Access*, vol. 6, pp. 39807–39819, 2018.

[25] X. Li, X. Tang, C.-C. Wang, and X. Lin, "Gibbs-sampling-based optimization for the deployment of small cells in 3G heterogeneous networks," in *Proc. 11th Int. Symp. Workshops Modeling Optim. Mobile, Ad Hoc Wireless Netw. (WiOpt)*, 2013, pp. 444–451.

[26] R. Götz, "Radio network planning tools basics, practical examples & demonstration on NGN network planning part I—Regional seminar on evolving network infrastructures to NGN and related planning strategies and tools, for the CEE, CIS and baltic states," LS telcom, Lichtenau, Germany, 2005. [Online]. Available: https://www.itu.int/ITU-D/tech/OLD_TND_WEBSITE/network-infrastructure_OLD/Belgrade-05/Day%204/Belgrade_6_5.pdf

[27] TEOCO RAN Solutions. (2019). *Planning Optimisation*. [Online]. Available: https://www.teoco.com/products/planning-optimization/

[28] Intermap Telecommunications. (2019). *Nextwave Telecom*. [Online]. Available: https://www.intermap.com/telecommunications

[29] P. Medeđović, M. Veletić, and V. Z. Blagojević, "Wireless insite software verification via analysis and comparison of simulation and measurement results," in *Proc. 35th Int. Conv. MIPRO*, 2012, pp. 776–781.

[30] Remcom. (2020). *Wireless Insite 3D Wireless Prediction Software*. [Online]. Available: https://www.remcom.com/wireless-insite-em-propagation-software

[31] P. Rice, A. Longley, K. Norton, and A. Barsis, "Transmission loss predictions for tropospheric communication circuits, volume 1," Inst. Telecommun. Sci. Aeronomy, Boulder, CO, USA, Tech. Rep. AD0687820, 1967. [Online]. Available: https://apps.dtic.mil/sti/pdfs/AD0687820.pdf

[32] A. Longley and L. R. Patricia, "Prediction of tropospheric radio transmission loss over irregular terrain," Tropospheric Telecommun. Lab., Inst. Telecommun. Sci., Boulder, CO, USA, Tech. Rep. AD0676874, 1968. [Online]. Available: https://apps.dtic.mil/sti/pdfs/AD0676874.pdf

[33] A. G. Longley, "Radio propagation in urban areas," in *Proc. 28th IEEE Veh. Technol. Conf.*, vol. 28, Mar. 1978, pp. 503–511.

[34] R. Edwards and J. Durkin, "Computer prediction of service areas for VHF mobile radio networks," *Proc. Inst. Elect. Eng.*, vol. 116, no. 9, pp. 1493–1500, 1969.

[35] C. E. Dadson, J. Durkin, and R. E. Martin, "Computer prediction of field strength in the planning of radio systems," *IEEE Trans. Veh. Technol.*, vol. VT-24, no. 1, pp. 1–8, Feb. 1975.

[36] Y. Okumura, E. Ohmori, T. Kawano, and K. Fukuda, "Field strength and its variability in VHF and UHF land-mobile radio service," *Rev. Elect. Commun. Lab.*, vol. 16, nos. 9–10, pp. 825–873, 1968.

[37] M. Hatay, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Technol.*, vol. VT-29, no. 3, pp. 317–325, Aug. 1980.

[38] E. Damosso and L. Correia, *Urban Transmission Loss Models for Mobile Radio in the 900 and 1800 MHz Bands*. Bassendean, NSW, Australia: The Hague, Sep. 1991.

[39] M. Brazil, R. L. Graham, D. A. Thomas, and M. Zachariasen, "On the history of the Euclidean Steiner tree problem," *Arch. History Exact Sci.*, vol. 68, no. 3, pp. 327–354, 2014.

[40] C. Tsallis, "What are the numbers that experiments provide," *Quimica Nova*, vol. 17, no. 6, pp. 468–471, 1994.

[41] P. Verma, A. Kumar, N. Rathod, P. Jain, S. Mallikarjun, R. Subramanian, B. Amrutur, M. S. M. Kumar, and R. Sundaresan, "Towards an IoT based water management system for a campus," in *Proc. IEEE 1st Int. Smart Cities Conf. (ISC2)*, Oct. 2015, pp. 1–6.

[42] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "Lorawan specification, version 1.1," LoRa Alliance, Oct. 2017, p. 101. [Online]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_-v1.1.pdf

[43] N. Rathod and R. Subramanian. (2020). *Relays and Coverage Repository*. [Online]. Available: https://github.com/nihesh7391/RelaysAndCoverage.git

[44] N. Rathod and R. Subramanian. (2020). *Experimental Data*. [Online]. Available: https://www.dropbox.com/sh/wdfut7lfobzi1km/AAA4xlfe7UKDoXgYy9IJVCoTa?dl=0

[45] N. Rathod and R. Subramanian. (2020). *Experimental Data*. [Online]. Available: https://drive.google.com/drive/folders/1byeVtzJ1xyJY0kXJm5i9ZM1esVMxozIx?usp=sharing

NIHESH RATHOD received the B.E. degree in electronics and communication from Dharmsinh Desai University, Nadiad, in 2012, and the M.E. degree in telecommunication from the Indian Institute of Science, Bengaluru, in 2015, where he is currently pursuing the Ph.D. degree in communication and networks. Since 2015, he has been a Cisco Research Scholar with the Department of Electrical Communication Engineering, Indian Institute of Science. His research interests include the areas in design and implementation of the Internet of Things (IoT).

RAJESH SUNDARESAN (Senior Member, IEEE) received the B.Tech. degree in electronics and communication from the Indian Institute of Technology Madras, India, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA, in 1996 and 1999, respectively. From 1999 to 2005, he worked at Qualcomm Inc. on the design of communication algorithms for wireless modems. Since 2005, he has been with the Indian Institute of Science, Bengaluru, India, where he is currently a Professor with the Department of Electrical Communication Engineering and an Associate Faculty with the Robert Bosch Centre for Cyber-Physical Systems. His research interests include the areas of communication, computation, and control over networks. He was an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY, from 2012 to 2015.