International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

IMPLEMENTATION OF UNMANNED AERIAL VEHICLES AS FLYING BASE STATIONS TO ASSIST 5G NETWORKS

KAKAULA RAMESHWARAMMA[1], Dr. N MAGESWARI[2]
[1]DECS 202T1D3806, ASHOKA WOMENS ENGINEERING COLLEGE, KURNOOL, A.P.
[2] Professor – ASHOKA WOMENS ENGINEERING COLLEGE, KURNOOL, A.P

ABSTRACT
Current wireless communication networks are not able to accommodate the increase in broadband data and are currently encountering fundamental challenges like higher data rate and Quality of Service (QoS) requirements, energy efficiency and excellent end-to-end performance and user coverage in overcrowded areas and hotspots whilst maintaining extremely low latency and high bandwidth. The deployment of 5G networks aims to address such challenges by introducing multiple advancements to the network and implementing new technologies to evolve new radio networks. This will primarily be done by introducing the 5G New Radio, which is the radio technology that is being developed to support the 5G technologies that will solve the problems mentioned previously. With the New Radio implementation, the next generation networks will accommodate the growing data rates. The networks are expected to attain a mobile data volume per unit area that is 1,000 times higher than current networks. Over 10-100 times the number of current connected devices is expected to be accommodated by 5G networks. Coverage is primarily the crucial problem with 5G networks, requiring the densification of urban areas with heterogeneous networks and the deployment of more closely packed terrestrial MBSs. However, this is not cost-effective and can be more complex as terrestrial network replanning will be required. The issue can be overcome by integrating UAVs into the network infrastructure as FBSs.
Key Words - *UAV,FBS, 5G,QoS.*

Introduction

The complementation of different technologies harvested by the Fourth Industrial Revolution is key to leaping towards the fully automated world of tomorrow. For such reasons, mobile data networks have become an integral and underlying structure of the world. The initial evolution of wireless networks was primarily a virtually infinite network of computers, key to the advancements of the 20th century and the restructuring of its economies [1]. The evolution of the internet brought by the advent of Third Generation wireless networks (3G) initiated the mobile internet, introducing the connectivity of devices. Fourth Generation wireless networks (4G) satisfied the growing thirst of the world to use the internet even more dynamically. This advent witnessed the emergence of the Internet of Things (IoT), which is currently connect- ing a number of devices over six times times larger than the global human population. Every advancement introduces newer network complexities and challenges such as higher data rate requirements and the limitation of available frequency spectrum. The existing wireless sys- tems will not be able to handle the exponential increases in mobile broadband data accelerated by the increased Machine-to-Machine (M2M) communications [8]. The much anticipated de- ployment of the Fifth Generation of wireless networks (5G) is expected to accommodate the increased data rates and requirements by providing larger scale connectivity. This will be done by enhancing current network technologies to improve the networks, and by introducing new technologies that will evolve a new network with unique standards. However, as previously mentioned, every advancement in the networks introduces more complex challenges. 5G net- works introduce a new spectrum range that offers extreme capacity and speed, but this comes at the expense of coverage. The high frequency waves used in the upper bound of the spectrum range that 5G employs travel at lower wavelengths, introducing the need for densification of the networks. More Terrestrial Macro Base Stations (MBSs) will need to be deployed to provide connectivity, with small cells dominating the urban areas of cities to complement the Macro Cell layer. However, due to

DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

geographical constraints and cost limitations, many regions will not be able to smoothly deploy the higher band spectrum of 5G networks. Recently, the deployment of Unmanned Aerial Vehicles (UAVs) in wireless networks has received significant attention and research to approach such constraints. The flexibility of UAV networks can over- come geographical constraints and cost limitations if deployed optimally. Developments such as Google's Project Loon and projects aiming to enable wide-scale communications through UAV employment by ATT and Qualcomm are motivating further research into the deployment of UAVs to act as Flying Base Stations (FBSs) [5] . FBSs can augment the terrestrial 5G network by enhancing operations in a plethora of varying use cases such as data collection for IoTs, in- formation dissemination, and machine-type communications. The FBSs can be used to support high communication demand areas such as large events or highly dense regions. In such use case scenarios, the UAV assisted 5G networks can provide seamlessly ubiquitous connectivity at any location, potentially well beyond the coverage area of a single terrestrial MBS. This will improve the network efficiency and flexibility and thus ease the growth in high data demand.

1. Literature Survey

In paper [1] the path optimization for flying base station is discussed and implemented.Paper [2] addressed the minimization of total Rotary-wing UAV consumption for wireless com- munication. The paper addressed different types of energy consumption such as propulsion, hovering, and communication related energy consumption. The paper succeeded in deriving a propulsion power consumption model which helped minimize the total UAV energy consump- tion. Traveling Salesman Problem (TSP)and convex optimization techniques were effectively used to demonstrate optimal hovering locations for the UAVs while still satisfying target GN communication throughput. An algorithm was proposed to solve the modelled optimization problems from the power consumption model, and the algorithm produced results which min- imized energy consumption for the UAVs. The algorithm significantly outperformed other approaches discussed in the paper such as the geometric center fly hover approach or the GN hover approach. Paper [2] weighed multiple approaches to solving the TSP for the trajectory optimization of UAVs, and the results displayed the superiority of the 2-opt NN algorithm. This motivated further investigation into using the algorithm in this project. The algorithm outperformed the PSO and GA heuristics both in terms of computational time and distance in some experimental results displayed in the paper. The paper also discussed the K-means clustering algorithm and other clustering approaches to be integrated with a TSP solver. This was also a source of motivation to implement some of the techniques in this project to opti- mize trajectory. The TSP was also largely investigated in papers [3] [4] [5]. Some of the papers provided results of algorithm experiments which were used to evaluate algorithms in this project. It was unknown whether the 2-opt algorithm could run exponentially in a special case of point distribution, until paper [6] addressed this issue by presenting such distributions. The below table summarizes some significant topics in FBS optimization by highlighting the papers addressing each problem.

Implementation

Clustering is the objective of grouping objects into multiple clusters based on the similarities between the objects. An efficient implementation of clustering will result in different clusters in which the objects have similarities that are different to those of objects in other clusters.[wiki clustering]. To solve the problem faced by the two FBSs, (3.5) will present an algorithm that will be devised to cluster the GNs using a local search technique. This will assist in splitting the two GN clusters according to the locations of the GNs in space. By using the local search techniques the clustering will focus on achieving the best possible multi-cell partitioning so that the MTSP problem faced by the FBSs is solved. In the standard MTSP, $m \geq 1$ salesmen must leave from one depot and return to it after having completed m tours subject to each tour passing through a city only once. Every city mast have been part of a single tour. The objective is that the resulting cost which is the addition of the distances

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

covered by the tours is the lowest cost possible. Consequently, the tours must be optimally directed. The variation faced by the FBSs differs from the standard MTSP because in the case of the FBSs, each FBS must return to the MBS that the FBS was appointed to. Hence, rather than all tours having to end at a single depot, each tour can end in a different depot subject to every tour returning to the depot it originated from. To assess the performance of the proposed algorithm, the results will be compared to those produced by similarly integrating the local search algorithm with the traditionally used K- means clustering algorithm. K-means clustering is effective in clustering object sets with using a technique that divides data sets based on centroids.

To efficiently cluster all GNs as previously mentioned, some algorithms will be proposed and analysed. The simulations that intend to find the optimal cell edge location by implementing the algorithms will then be explained.

Summary of Traditional Algorithms

As previously discussed, the 2-opt local search applied to a NN tour will be used in the algo- rithms that will be proposed. The 2-opt NN algorithm follows the

Algorithm 1 2-opt NN

Input: 1. (n points) 2. (xy coordinates of points) 3.(Starting NN tour

Output: 1. Tour between n points

```
Evaluate the distance matrix
Define the NN tour
for i = 1 : n − 2 do
    for j = i + 2 : n do
        evaluate total length d1 of two edges
        evaluate total length d2 of two edges when edges are swapped according to a 2-opt move
        d2 < d1 implement edge swap
    end
end
```

K-means Clustering will be used as the standard of evaluation of the proposed algorithms to verify the efficiencies and performances of the algorithms. K-means uses the spatial distri- bution of n points to partition the points in to k clusters. The algorithm works by randomizing locations of k centroids and calculating the distances between every point and every centroid, assigning points to the clusters centered at the centroids closest to the points [2]. The algo- rithm then replicates new centroids located at the centres of gravities of the points assigned to each cluster. The algorithm completes once the centroids are located at the mean of all points in the cluster assigned to the centroid, and no more changes can be made. K-means is known to have a complexity O(n2) The pseudocode of the algorithm is below [3].

Algorithm 2 K-means Clustering

Input: 1. (k - the number of clusters) 2. (D - a set of lift ratios)

Output: 1. Set of k clusters

Method:

Arbitrarily choose k centroids as inital cluster centers

Repeat:

1. assign each data point to the cluster with the closest centroid
2. update the centroid locations by finding the new mean values of the clusters

Until: no more updates possible

Analysis of Proposed Algorithms

Consider a two dimensional coordinate plane $P \subset \mathbb{R}^2$ in Euclidean space and two coordinates x, y where the point $(x, y) \in P$ where x and y are the largest values on the X-axis and the Y-axis respectively. The area size of P is x × y and there is a distribution of n points on the plane where each point

$$p \in \{p_1, p_2, \ldots, p_n\} \text{ and } \{\forall p \exists! (x_p, y_p) | (0 \leq x_p \leq x) \wedge (0 \leq y_p \leq y)\}$$

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

. Both x and y in the following algorithms are equal to the Euclidean distance between two terrestrial MBSs.

Algorithm 1: FBS Linear Boundary-search

Algorithm 1 intends to find the optimal location of the linear cell edge that separates two cells by finding the net distance covered, using the 2-opt NN algorithm, by each FBS at multiple cell edge locations. The output will be the cell edge location at which the net distance is the minimum net distance. This will group the GNs within the cells into two clusters each of which will be serviced by the FBS appointed to the terrestrial MBS in the according cell. Algorithm 1 iterates through 7 X-coordinates representing a linear boundary $b_q \in \{b_1, b_2, ..., b_7\}$ where bq is perpendicular to the X-axis at $x_{qi} | \frac{7x}{20} \leq x_{qi} \leq \frac{13x}{20}$. The algorithm does so by incrementally increasing the value of qi within the range $(\frac{7x}{20}, \frac{13x}{20})$ by $\frac{x}{20}$ at every iteration.

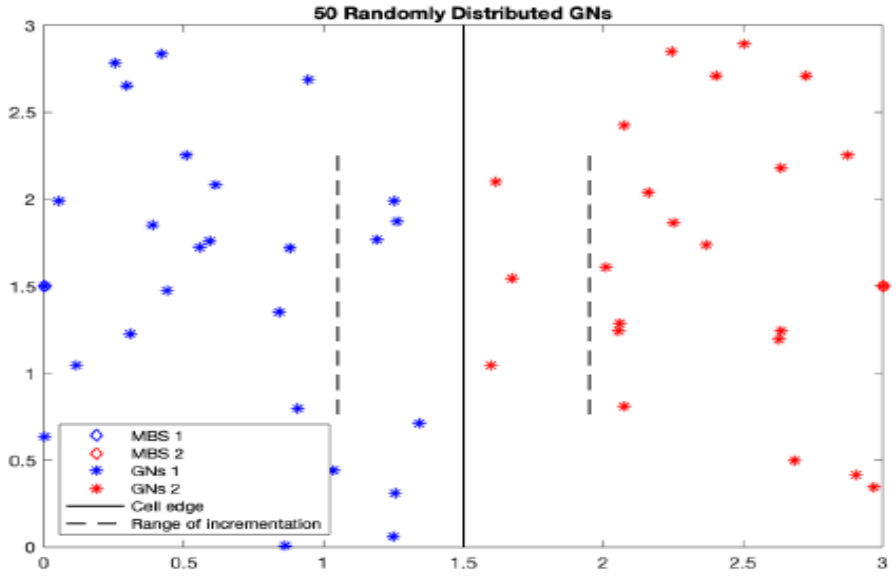


Figure 1: Boundary Incrementation Range

Figure 1 can be used to visualize how the algorithm will run.
The net distance covered bythe FBSs will be calculated at seven locations of the boundary. The first boundary is located at the dashed line on the left and the last at the dashed line on the right An explanation of thealgorithm inputs and outputs is below followed by the pseudocode on the following page.

Input:
Is the value of the distance (km) between the two terrestrial MBSs.
Is the number of GNs to be served by the FBSs.
Is input as a 1 × 2 matrix [power(W) velocity(m/s)].

Output:
Is a 1 × 7 matrix containing the net distance (km) covered by the FBSs at every boundary location.
Is a 1×7 matrix containing the energy (J) consumed by FBS 1 at every boundary location.
Is a 1×7 matrix containing the energy (J) consumed by FBS 2 at every boundary location.
Is a 2 × 2 matrix containing the coordinates of the boundary location with the least net distance min(1) such that if the matrix was plotted with a line intersecting the two pointsthe boundary would be the output.
Is the X-coordinate of the optimal boundary location.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

Algorithm 3 FBS_Boundary_search_lin
Input: 1. (cell_diamater) 2. (GNs) 3. (UAV_parameters)
Output: 1. (dnet_lin) 2. (ec1_lin) 3. (ec2_lin) 4. (coordinates_optimal_edge_lin) 5. (x_optimal_edge_lin)

```
Add MBS coordinates to GNs matrix
for B = 1:7 do
    Increment X-coordinate of boundary
    Assign points left of new boundary
    Implement 2-opt search to the assigned points
    Find energy consumed by FBS 1
    X = M(x_M>x_boundary(1),:);%Build matrix of points right of new boundary
    Assign points right of new boundary
    Implement 2-opt search to the assigned points
    Find energy consumed by FBS 2
end
Output optimal edge index
return Energy consumption of FBS 1,2 and Net distance and Optimal boundary coordinates
```

The 2-opt algorithm firstly implements the greedy NN algorithm, which has a computational complexity of O(n2), to find a starting tour. The 2-opt algorithm, which also has a complexity of O(n2), is then implemented to improve the initial tour. As observable in the 2-opt NN pseudocode, the 2-opt begins running after NN runs, and so the exact worst case running time is (2 × n2). The constant 2 will be disregarded as n → ∞ in the worst case. Therefore, the computational complexity of the 2-opt NN algorithm is O(n2). As observable in the pseudocode of Algorithm 1, the 2-opt runs simultaneously within the boundary iteration loop. The loop runs from B = 1:7, hence, the exact worst case running time is (7 × n2). Considering a constant upper bound of 7 for B, the constant 7 will be disregarded as n → ∞ in the worst case. Therefore, the underlying computational complexity of Algorithm 1 , where n is the size of GNs, is O(n2). The algorithm runs in polynomial time, making the implementation quick.

Algorithm 2: FBS Piece-wise Boundary-search

Algorithm 2 intends to further optimize the cell edge location by improving the output of Algorithm 1. The algorithm does so by inputting the optimal linear cell edge location and outputting an improved piecewise cell edge, further optimizing the GNs clustering. The same general procedure that Algorithm 1 follows is followed by Algorithm 2. The algorithm does soby initially using the output from Algorithm 1: the X-coordinate of the optimal linear cell edgelocation. Algorithm 2 then splits the linear boundary at two points p_2 and p_3 located at $(\frac{x}{2}, \frac{3y}{4})$ and $(\frac{x}{2}, \frac{1y}{4})$ respectively.
Algorithm 2 then iterates through five different locations of p2 and does so for p3 at every location of p2. Both p2 and p3 are incremented within the range $(\frac{7x}{20}, \frac{13x}{20})$as observed in Figure 3.6 below. Algorithm 2 will then iterate through 25 different piecewise boundary locations by incrementally increasing each point by x/40 and output the boundary at which the net distance covered by the FBSs is the lowest.

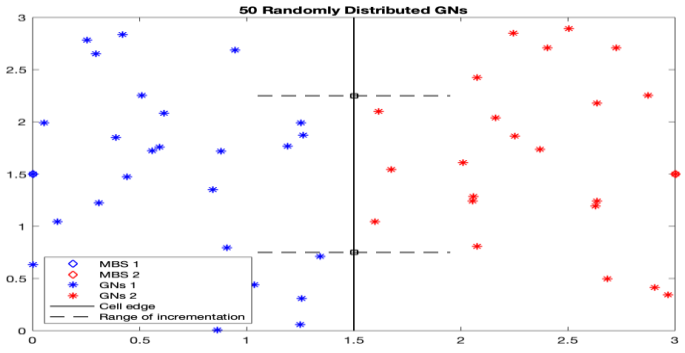


Figure 2: Point Incrementation Range

Input:

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

Is the value of the distance (km) between the two terrestrial MBSs.

Is the number of GNs to be served by the FBSs.

Is input as a 1 × 2 matrix [power(W) velocity(m/s)].
Is the X-coordinate of the optimal linear boundary location.

Output:
Is a 1 × 7 matrix containing the net distance (km) covered by the FBSs at every piecewise boundary location.
Is a 1×7 matrix containing the energy (J) consumed by FBS 1 at every piecewise boundary location.
Is a 1×7 matrix containing the energy (J) consumed by FBS 2 at every piecewise boundary location.
Is a 4 × 2 matrix containing the coordinates of points at the starting point, endpoint, and the points at which the piecewise boundary with the least net distance min(1) splits such that if the matrix was plotted with a line intersecting the four points the piecewise boundary would be the output.

Algorithm 4 FBS_Boundary_search_pw
Input: 1. (cell_diamater) 2. (GNs) 3. (UAV_parameters) 4. (x_optimal_edge_lin)
Output: 1. (dnet_pw) 2. (ec1_pw) 3. (ec2_pw) 4. (coordinates_optimal_edge_pw)
Add MBS coordinates to GNs matrix
for *p2 = 1:5* **do**
  Increment X-coordinates of p2
  for *p3=1.5* **do**
    Increment X-coordinates of p3
    Build a polygon bounded by piecewise boundary enclosing area left of piecewise boundary;
    Implement 2-opt on points inside the polygon
    Find energy consumed by FBS 1
    Implement 2-opt on points not inside the polygon
    Find energy consumed by FBS 2
    Group all net distances in matrix
  end
end
Find optimal edge index
return Energy consumption of FBS 1,2 and Net distance and Optimal boundary coordinates

As previously discussed, the computational complexity of the 2-opt NN algorithm is O(n2). As observable in the pseudocode of Algorithm 2, the tspsearch runs simultaneously within the p3 iteration loop. The p3 iteration loop runs simultaneously within the p2 iteration loop. Both loops p2 and p3 run from p2, p3 = 1:5, hence, the exact worst case running time is (25 × n2). Considering a constant upper bound of 25 for p2 × p3, the constant 25 will be disregarded as n → ∞ in the worst case . Therefore, the underlying computational complexity of Algorithm 2 , where n is the size of GNs, is O(n2). The algorithm runs in polynomial time, making the implementation quick.

Algorithm 3: FBS Boundary-search

Algorithm 3 combines both Algorithms 1,2 to efficiently find the optimal piecewise edge with only one input layer. The algorithm uses the input of Algorithm 1 and outputs the output of Algorithm 2. The pseudocode is as follows.

Algorithm 5 FBS Boundary search
Input: 1. (cell diamater) 2. (GNs) 3. (UAV parameters)
Output: 1. (dnet pw) 2. (ec1 pw) 3. (ec2 pw) 4. (coordinates optimal edge pw)
41 Implement Algorithm 1 to find optimal linear edge
42 Implement Algorithm 2 to find optimal piecewise edge
43 return Energy consumption of FBS 1,2 and Net distance and Optimal boundary coordinates

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

As observable in the pseudocode above, Algorithm 2 begins running after Algorithm 1 runs, and so the exact worst case running time is (2n2). The constant 2 will be disregarded as n → ∞ in the worst case. Therefore, the underlying computational complexity of Algorithm 3 , where n is the size of GNs, is O(n2). The algorithm runs in polynomial time, making the implementation quick. The parameters of Simulation 1 are summarized in the following table.

Simulation 1 Parameters

Parameter	Description	Value
MC	Monte Carlo runs	1000
GNs	Ground Nodes	20:10:80
P(W)	UAV Power	50
V(m/s)	UAV Velocity	60
A(km^2)	(Distance between MBSs)^2	3×3
U(a,b)	Uniform Distribution of GNs	

RESULTS

In Simulation 1 the objective was to find the optimal location of a linear cell edge separating two adjacent cells. This was done to optimize the FBS trajectory and minimize the associated costs. The results of Simulation 1 are as follows.

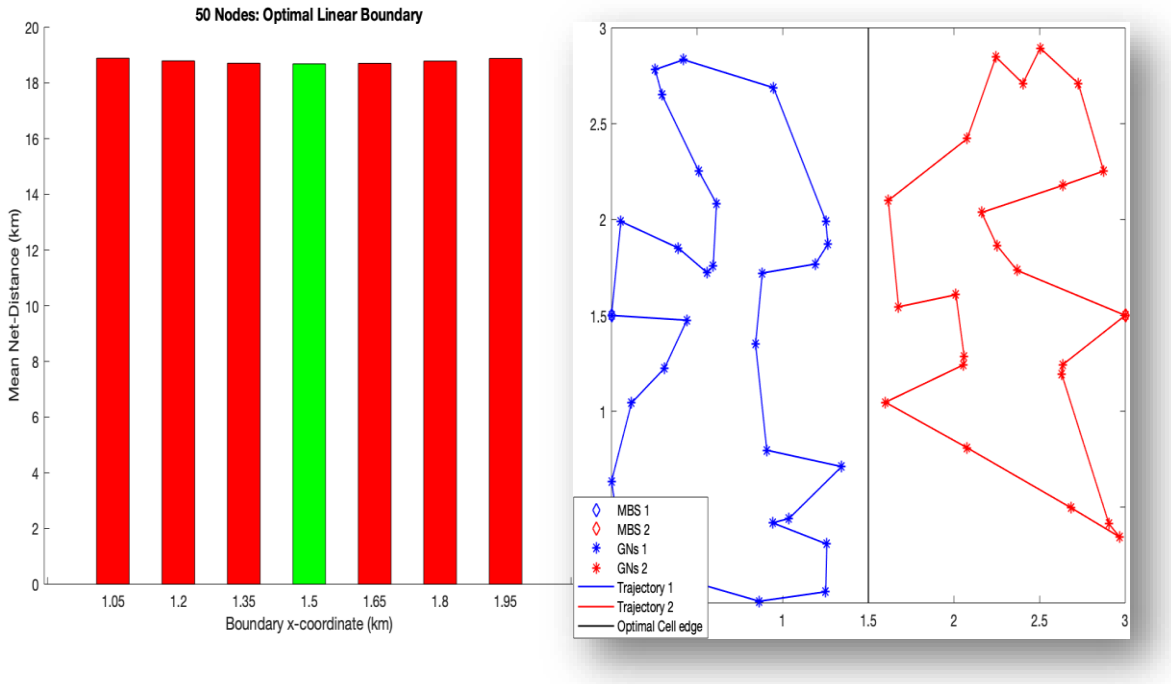


Figure 3: Optimal Linear Cell Edge

Figure 4: Optimal Linear Cell Edge for 50 GNs - Instance 1 Trajectory Representation

Location after 1,000 Instances

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

As expected and observed in the above figures, when uniformly distributing GNs the central boundary is the optimal location for the linear cell edge as the lowest mean net distance is at the boundary with an X-coordinate of 1.5km in the 3km axis.. However, Figure (4.2) displays a pattern of decreasing values from left to center. Within boundaries 3 and 5 there is a boundary location at which the mean net distances begin increasing and the results display a continual increase until the final location. Note that the exact location that has the lowest possible mean net distance, precisely before the increase of values begins, is the optimal location for the boundary. To further investigate the area, Simulation 2 will present the results of the optimal piecewise boundary, an extension of boundary 4 (1.5km) in Figure (4), within boundaries 3 and 5. The results of all other GN sets are the same as the above result and follow the same pattern discussed, and will therefore be displayed in a table below with the standard deviations of results of every node set.

Key to Table (1):

Is the X-coordinate of the point (x,0) at which the optimal linear boundary intersects the X-axis.

Is the mean net distance of all 1,000 instances at the optimal boundary location.

Is the standard deviation of the data set containing the seven mean net distances at each boundary obtained after all instances for all GN sizes. An example of a data set is the data set presented in Figure (4.1) for 50 GNs.

Is the value of the standard deviation of the data set containing the 1,000 net distances of the boundary that had the highest standard deviation in the instances.

Table 2: Simulation 1 Summary of Results

GNs	20	30	40	50	60	70	80
(1) x(km)	1.5	1.5	1.5	1.5	1.5	1.5	1.5
(2) μ_l(km)	13.33	15.41	17.1	18.69	20.05	21.35	22.58
(3) σ_1(km)	0.0303	0.0720	0.0926	0.0829	0.0806	0.0601	0.0610
(4) σ_2(km)	1.1420	1.0426	0.9382	0.8787	0.8368	0.8431	0.8352

The results in the above table summarize all findings from implementing Algorithm 1 in Simulation 1 to find the most optimal linear cell edge location. The central boundary at 1.5km was found to be the optimal location for the linear boundary. The largest of seven boundary standard deviations of 1,000 instances was recorded for 20 GNs. The GNs size and the reliability of the results are directly proportional as with the increase of GNs size, the table shows a decrease in the value of the worst standard deviation of 1,000 runs with the exception of a discrepancy in σ2 of 70 GNs.

Results: Simulation 2

In Simulation 2 the objective was to improve on the results of Simulation 1 by focusing on the search space within the three most optimal linear boundary locations. This was done by converting the optimal linear boundary into a three-segment piecewise boundary and obtaining the results of multiple piecewise boundaries within the optimal region in the X-coordinate range (1.35,1.65)km. This will further optimize the FBS trajectory and minimize the associated costs. The results of Simulation 2 are as follows.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

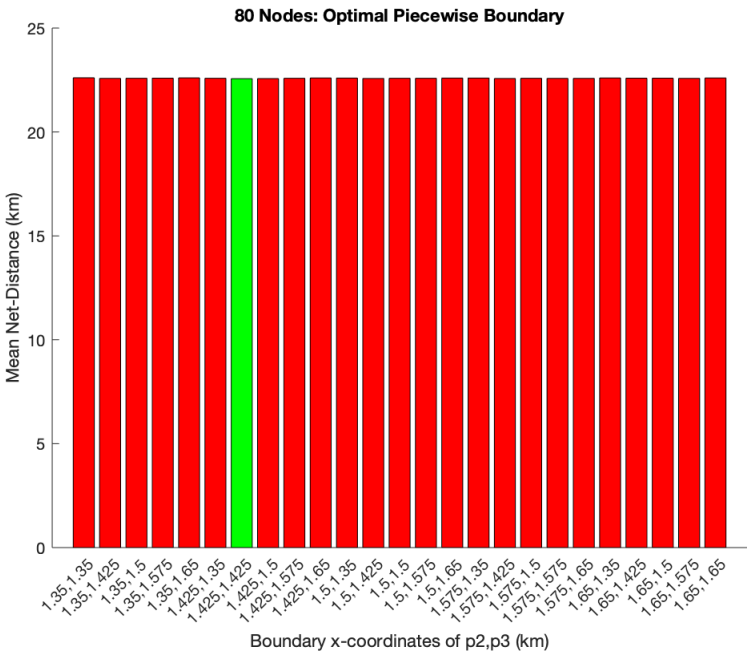


Figure 5: Optimal Piecewise Cell Edge Location after 1,000 Instances

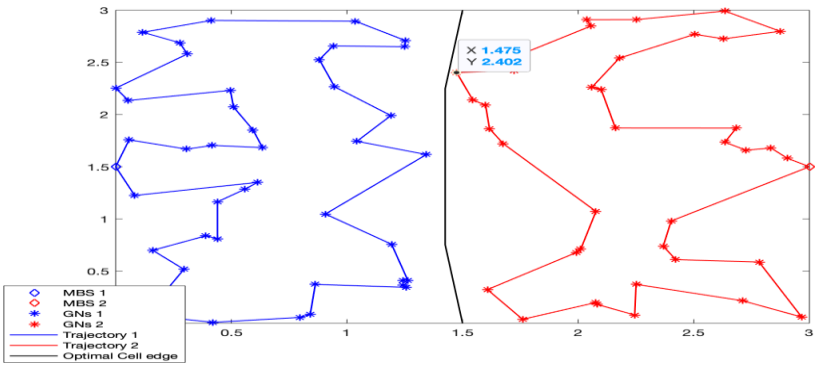
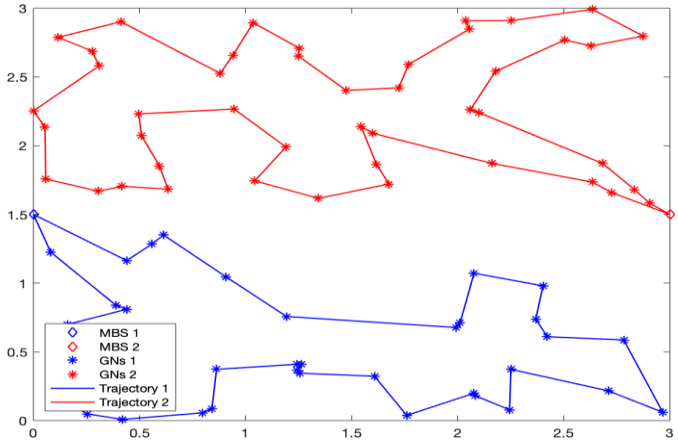


Figure 6: Optimal Piecewise Cell Edge for 80 GNs - Instance 1 Trajectory Representation

As observed in the above figures, when uniformly distributing GNs a piecewise cell edge is most optimal at coordinates [(1.5,3) (1.425,2.25) (1.425, 0.75) (1.5,0)]. The piecewise edge improved the net distance by approximately 20m. The most significant improvement was observable when experimenting with 80 nodes. This is expected as Algorithm 2 implements the search in a very



International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

specific area to improve the local optimum obtained by Algorithm 1. The higher the density of GNs, the larger the effect of Algorithm 2. The algorithm effectively changed the cluster of the point with highlighted coordinates in Figure (4.4) and hence obtained an improved optimum. The algorithm is especially effective when dealing with large distances and GN sizes. As previously mentioned when discussing the cell edge placement problem, a single GN can make a difference to the
Figure 7: K-means clustering of 80 GNs at Instance 1

trajectories of the FBSs. For these reasons, such algorithms can be implemented to reduce the costs by any means attainable. To test the clustering performance of Algorithm 2, a comparison of results to K-means clustering will be presented and discussed below.

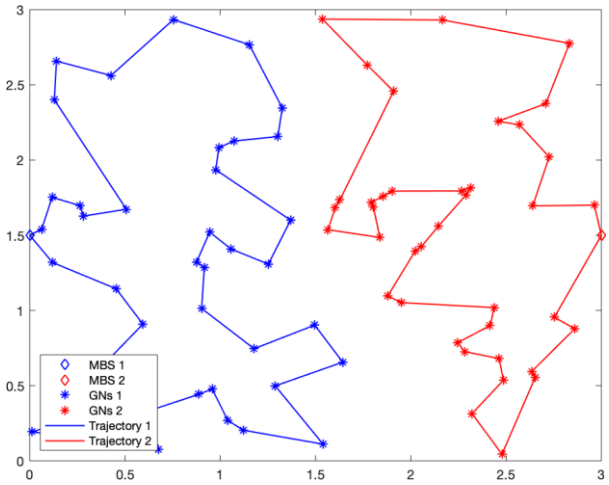


Figure 8: K-means clustering of 80 GNs at Instance 10

It is observable from the above figures that K-means clustering implements more dynami- cally than Algorithm 2. To assess the actual performances the below figures will compare the mean net distance of the optimal piecewise boundary obtained by Algorithm 2, to the mean net distance obtained by K-means clustering. The effect of the clustering methods on the energy consumption per FBS will also be assessed below. The above figure displays a superiority in the performance of Algorithm 2 over K-means clustering. For all GN sets, the lowest mean net distance obtained by Algorithm 2 is lower than the mean net distance obtained by K-means clustering.

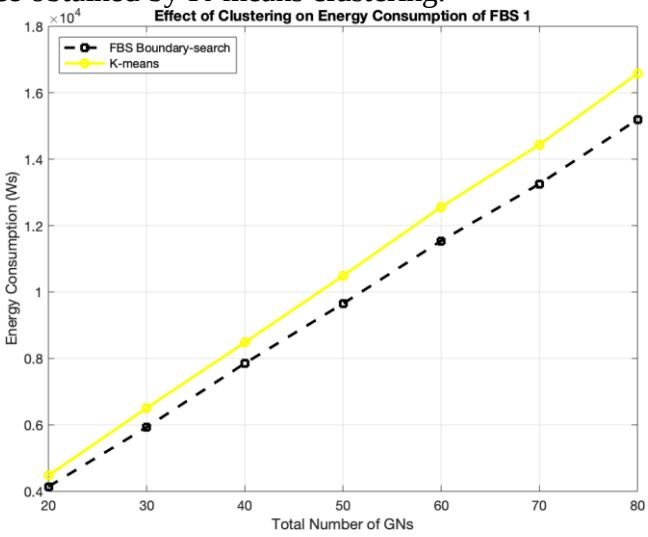


Figure 9: Effect of Clustering on Energy Consumption of FBS 1 after 1,000 Instances

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

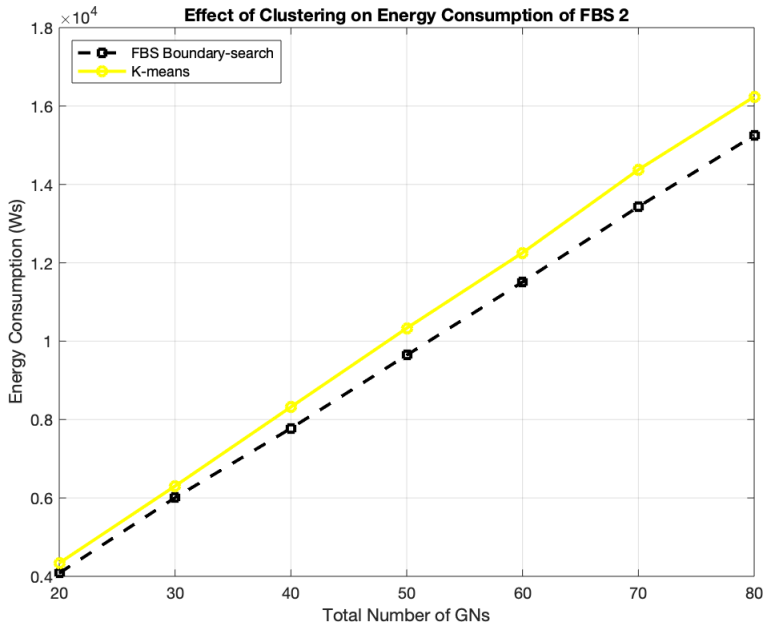


Figure 10: Effect of Clustering on Energy Consumption of FBS 2 after 1,000 Instances Both FBSs consume less energy when GNs are clustered by Algorithm 2 for every size of GNs. The algorithm maintains a lower slope on both graphs than K-means by 9% for FBS 1 and 2% for FBS 2, highlighting the superiority in the algorithm's effectiveness with increasingnumbers of GNs. The number of GNs increases the energy consumed as energy consumption of a rotary-wing UAV is not limited to propulsion energy due to other energy consuming qualities of the UAV such as hovering. In the results, both propulsion energy and hovering energy are considered. For these reasons, the number of GNs increases energy consumption of an FBS not only because of the increased distance, but also because of the increased GNs to hover above. However, the FBSs both incur the effect of larger node sizes with lower significance when clustered by Algorithm 2 than when clustered by K-means. Algorithm 2 also maintains a stronger balance and consistency in the clustering as the rate of change in energy consumedper change in GN size is 7% higher for FBS 1 than for FBS 2. However, in the case of K-meansclustering, the rate for FBS 1 surpasses the rate for FBS 2 by 14%. Algorithm 2 succeeds in minimizing effect of the GN size on the energy consumed, for both FBSs, when compared to K-means clustering. The algorithm does so by clustering more effectively and balancing the net FBS costs. The below table summarizes the results of Simulation 2.

Key to Table (3):

1. Is the X-coordinates of two points with coordinates (x_1,2.25) and (x_2,0.75)km at which the optimal piecewise boundary is incremented.
2. Is the mean net distance of all 1,000 instances at the optimal boundary location.

3. Is the decrease in distance from the output of Algorithm 1.

4. Is the standard deviation of the data set containing the 25 mean net distances at each boundary obtained after all instances for all GN sizes. An example of a data set is the data set presented in Figure (4.4) for 80 GNs.
5. Is the value of the standard deviation of the data set containing the 1,000 net distances of the boundary that had the highest standard deviation in the instances.
6. Is the mean net distance of all instances of the K-means clusters.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

Table 3: Simulation 2 Summary of Results

GNs	20	30	40	50	60	70	80
Algorithm 2							
(1) (x_1, x_2)(km)	(1.425,1.575)	(1.5,1.5)	(1.5,1.5)	(1.5,1.5)	(1.5,1.575)	(1.5,1.425)	(1.425,1.425)
(2) μ_p(km)	13.32	15.41	17.1	18.69	20.04	21.35	22.56
(3) $\mu_l - \mu_p$(m)	10	0	0	0	10	0	20
(4) σ_1(km)	0.0072	0.0110	0.0129	0.0135	0.0106	0.0096	0.0105
(5) σ_2(km)	1.1071	0.9651	0.9091	0.8759	0.8267	0.8531	0.8550
K-means Clustering							
(6) μ_k(km)	13.54	15.68	17.33	18.87	20.21	21.45	22.64
(7) σ_k(km)	1.2972	1.0905	0.9728	0.9225	0.8441	0.8186	0.8080

Conclusion

In the proposed work, the FBS search and K means are compared

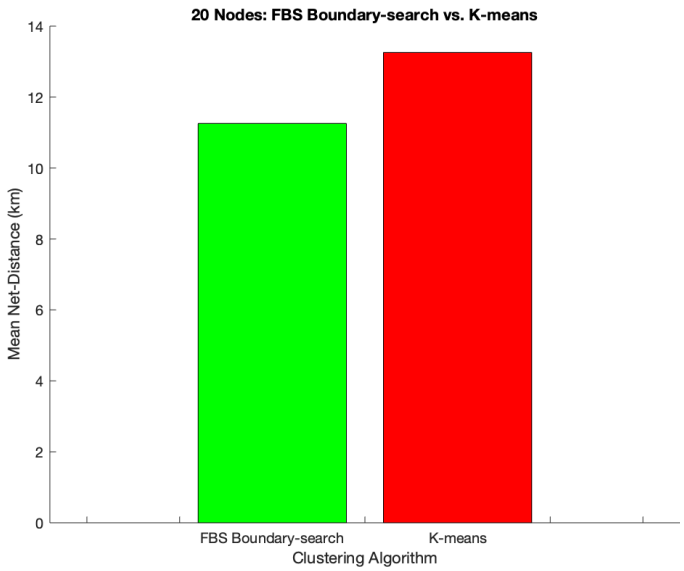


Figure 11: Comparison of Algorithm 2 and K-means Clustering - Mean Net Distance

Table 4: Summary of Results

GNs	20	30	40	50	60	70	80
Algorithm 2							

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

(1) (x_1, x_2)(km)	(2.1,2.1)	(1.425,1.35)	(1.2,1.275)	(1.125,1.125)	(1.05,1.125)	(1.05,1.125)	(1.125,1.30)
(2) μ_p(km)	11.26	13.34	14.81	16.09	17.19	18.21	19.23
(3) $\mu_l - \mu_p$(m)	30	0	30	10	10	50	50
(4) σ_1(km)	0.0222	0.0148	0.0258	0.0218	0.0291	0.0529	0.0522
(5) σ_2(km)	1.6593	1.3884	1.1829	1.1579	1.0798	1.1011	1.0966
K-means Clustering							
(6) μ_k(km)	13.26	15.14	16.56	17.88	18.89	19.93	20.86
(7) σ_k(km)	1.5594	1.361	1.1455	1.0719	0.9776	0.9601	0.9053 hei

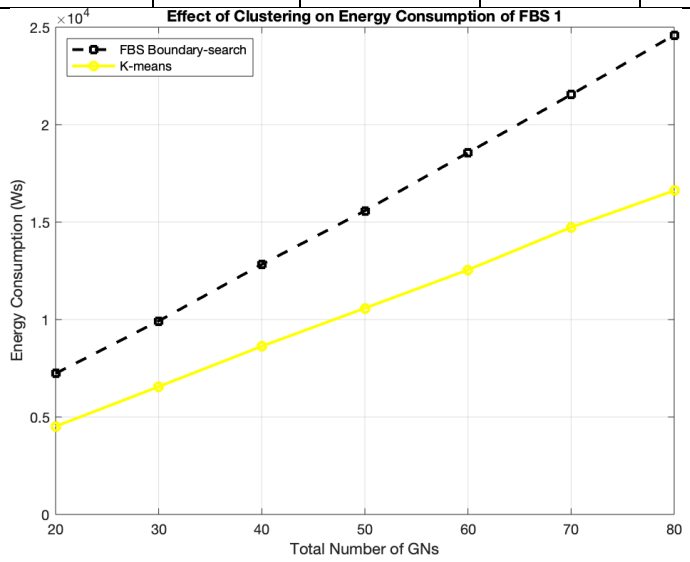
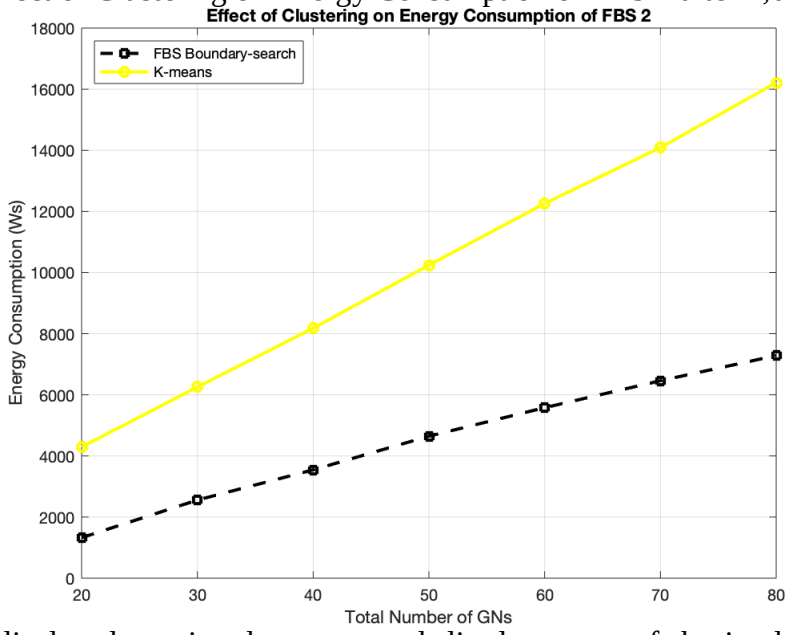


Figure 12: Effect of Clustering on Energy Consumption of FBS 1 after 1,000 Instances

Figure 13: Effect of Clustering on Energy Consumption of FBS 2 after 1,000 Instances



The above results display dynamic advantages and disadvantages of the implementation of each algorithm when dealing with the flash-crowd scenario. When assessing the energy consumption of

DOI:10.46647/ijetms.2023.v07i01.024 ISSN: 2581-4621

FBS 1, Algorithm 2 produces results that are significantly more detrimental to the FBS's energy consumption than K-means does. The algorithm also outputs a larger increase in energy consumption for every increase in GN size than K-means. This is due to theslope of Algorithm 2 surpassing that of K-means by approximately 62%. However, K-means only managed to achieve better FBS 1 results by assigning flash-crowd segments in cell 1 to FBS 2. As observed in Figure (4.24), Algorithm 2 maintains a steady slope lower than the steeper slope of K-means by approximately 158%, due to the output of deploying FBS 2 to assist users far from the flah-crowd. Also, the maximum energy consumed by FBS 1 is 17.3kJ higher than by FBS 2 for Algorithm 2. On the other hand, the difference for K-means is 410 J higher for FBS 1. However, Algorithm 2 maintains a lower net energy consumption of both FBSs. In conclusion, K-means succeeds in minimizing any overcrowded area by dividingthe servicing mission amongst two FBSs. The project addressed the potential of deploying UAVs as FBSs to assist 5G Networks. Trajectory optimization of FBSs was a key objective of this project, and hence intensive research on the TSP was discussed to better understand the problem and the approaches to solving it. After find an optimal route for a single FBS, a second problem arose: the optimal cell edge placement. This problem was dependent on the trajectories of both FBSs as the problem was primarily to group all GNs in the multi-cell space into two clusters so that each could be serviced by a FBS. The primary concern was when a flashcrowd overwhelms one cell, and this was the key motive of trying to find a more optimal location for the cell edge so that the GNs can be clustered based on the dispersion of users rather than the geographical locations of the terrestrial MBSs. An algorithm that finds an optimal linear cell edge location based on the user dispersion was proposed. A second algorithm that improves the results of the first was also proposed, and this algorithm successfully and consistently outperformed K-means clustering when tasked with clustering GNs the multi-cell space. Future works on the aforementioned topics could include the improvement of the algorithms so that more dynamic clusters can be found. This would be possible by rotating the cell edge at an angle at every iteration. This would then diversify the applications of the algorithms in future applications. The algorithms could also be used to cluster GNs within the same cell to determine the trajectories of multiple FBSs deployed at the same MBS. A further improvement that can be made to the experiments would be the application of K-means to find optimal hover locations, rather than hovering over all GNs. This would be significant in a flashcrowd event as the FBS would be able to serve denser crowd more easily from the same location. The limitations of the above works are firstly the algorithms, which did produce significant results against K-means, however, K-means is more dynamic and can be applied to a multitude of applications. The current algorithms are very focused on cell spaces and not generic clustering. Some variations can be made to develop the algorithms so that they could be more applicable.

References

[1] Jongyul Lee and Vasilis Friderikos, "Path Optimization for Flying Base Stations in Multi-Cell Networks",978-1-7281-3106-1/20©2020 IEEE

[2] Qingqing Wu, Yong Zeng, and Rui Zhang. Joint trajectory and communication design for multi-uav enabled wireless networks. IEEE Transactions on Wireless Communications, 17(3):2109–2121, 2018.

[3] Christos H Papadimitriou and Kenneth Steiglitz. On the complexity of local search for the traveling salesman problem. SIAM Journal on Computing, 6(1):76–83, 1977.

[4] David S Johnson and Lyle A McGeoch. Experimental analysis of heuristics for the stsp. In The traveling salesman problem and its variations, pages 369–443. Springer, 2007.

[5] Zolt´an A´d´am Mann.The top eight misconceptions about np-hardness.Computer, 50(5):72–79, 2017.

[6] Ekram Hossain, Mehdi Rasti, Hina Tabassum, and Amr Abdelnasser. Evolution toward 5g multi-tier cellular wireless networks: An interference management perspective. IEEE Wireless Communications, 21(3):118–127, 2014.

[7] Harpreet S Dhillon, Howard Huang, and Harish Viswanathan. Wide-area wireless commu- nication challenges for the internet of things. IEEE Communications Magazine, 55(2):168– 174, 2017.