

Environment-Aware Deployment of Wireless Drones Base Stations with Google Earth Simulator

Aaron French, Mohammad Mozaffari, Abdelrahman Eldosouky, and Walid Saad

Wireless@VT, Electrical and Computer Engineering Department, Virginia Tech, VA, USA,
Emails: {ajfrench, mmozaff, iv727, walids}@vt.edu.

Abstract— In this paper, a software-based simulator for the deployment of base station-equipped unmanned aerial vehicles (UAVs) in a cellular network is proposed. To this end, the Google Earth Engine platform and its included image processing functions are used to collect geospatial data and to identify obstacles that can disrupt the line-of-sight (LoS) communications between UAVs and ground users. Given such geographical information, three environment-aware optimal UAV deployment scenarios are investigated using the developed simulator. In the first scenario, the positions of UAVs are optimized such that the number of ground users covered by UAVs is maximized. In the second scenario, the minimum number of UAVs needed to provide full coverage for all ground users is determined. Finally, given the load requirements of the ground users, the total flight time (i.e., energy) that the UAVs need to completely serve the ground users is minimized. Simulation results using a real area of the Virginia Tech campus show that the proposed environment-aware drone deployment framework with Google Earth input significantly enhances the network performance in terms of coverage and energy consumption, compared to classical deployment approaches that do not exploit geographical information. In particular, the results show that the proposed approach yields a coverage enhancement by a factor of 2, and a 65% improvement in energy-efficiency. The results have also shown the existence of an optimal number of drones that leads to a maximum wireless coverage performance.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs), or drones have recently attracted significant attention as a promising approach to enhance wireless communication performance [1]–[5]. When equipped with wireless base station hardware, drones can supplement the coverage provided by existing cellular infrastructure. The mobility of drones facilitates the creation of line-of-sight (LoS) links with users, ensuring optimal connection strength. This ability, coupled with the reliability and autonomy of drones, lends UAVs attractive qualities to service providers. In particular, UAVs are an effective approach in emergency scenarios such as disaster relief, when unplanned power outages may compound with the increased need for communication, and Internet of Things (IoT) applications [6], where the quantity and low transmit power of devices may necessitate closer-ranged wireless communications. Meanwhile, UAVs can also be used to complement existing terrestrial cellular systems by bringing additional capacity to crowded areas during temporary events. Furthermore, drones can be

deployed to provide necessary wireless connectivity to rural areas in which the presence of large-scale ground wireless infrastructure is limited.

To effectively deploy drone base stations in wireless networking applications, there is a need for efficient simulators that can simulate different use-case scenarios and ground environments. Though many simulators have been developed for terrestrial base stations [7], [8], only some are suited specifically for the analysis of three-dimensional, ad hoc networks [9]. These are typically implemented as extensions of the general network simulators [10], that operate in two dimensions. UAV-enabled networks are highly dynamic and thus require a proper integration of the movement of UAVs into the simulation environment. Moreover, analysis of these networks is made more challenging by the uncertainty of environmental variables affecting propagation, as well as highly dynamic interference. To account for these UAV features, many models implement probabilistic expressions based on environment type, i.e., rural, urban, or dense urban [11]. Thus, the ability to identify obstacles by processing satellite images can have immense value in that drone simulations can become more deterministic, depending on the accuracy of image processing.

While there has been a notable number of works on UAV deployment, most of them ignored the potential use of real geographical information for optimal placement of UAVs. For instance, the authors in [12] studied the coverage maximization problem with minimum number of drone base stations. In [13], the deployment of an aerial UAV base station for maximizing sum-rate and power gain in a wireless network is studied. The work in [14] studied the trajectory design for drone BSs while using a 3D city map information. These studies use variations of the probabilistic models introduced above, and thus, are not suited for simulation of real-world environments. In contrast to previous studies on UAV deployment, we extract environmental information with great precision by using image processing tools in Google Earth Engine. Subsequently, we build a drone deployment simulator that accepts buildings' locations as inputs, and adaptively determines the optimal positions of the drones for maximizing wireless connectivity in various scenarios.

The main contribution of this paper is a novel simulation framework for *environment-aware* deployment of multiple drone base stations that provide wireless connectivity for ground users. In particular, by exploiting geographical information extracted from the Google Earth Engine, we

This research was supported by the U.S. National Science Foundation under Grants CNS-1460316 and IIS-1633363.

determine the locations of buildings that disrupt LoS. Then, we use our simulator to investigate three key UAV deployment scenarios. First, we study the optimal placement of drones for maximizing the number of covered ground users. In the second scenario, we aim to provide full coverage for ground users by using a minimum number of drones. Finally, given the load requirements of users, we analyze the optimal deployment of drones for which the total flight time of drones needed to service the users is minimized. Simulation results reveal that our proposed framework that exploit buildings' information on an area in Virginia Tech's campus using Google Earth yields a significant improvement in the coverage and energy efficiency of the drone-enabled wireless networks. Moreover, our results show the existence of an optimal number of drones that maximizes the wireless connectivity.

The rest of this paper is organized as follows. In Section II, we present the system model the drone deployment scenarios. In Section III, we describe the developed feature (i.e., obstacle) extraction method from Google Earth. Simulation results are presented in Section IV and conclusions are drawn in Section V.

II. SYSTEM MODEL AND DRONE DEPLOYMENT SCENARIOS

Consider a set \mathcal{L} of L single-antenna wireless users located within a given geographical area. The location of a user $i \in \mathcal{L}$ is given by (x_i, y_i) . In this area, a set \mathcal{M} of M quadrotor drones are used as flying base stations to provide downlink wireless service to ground users, as shown in Figure 1. The location of a drone $j \in \mathcal{M}$ is given by $\mathbf{v}_j = (x_j^D, y_j^D, h_j)$.

Each user i can be served by one drone j that provides the strongest downlink signal-to-interference-plus-noise-ratio (SINR) for the user such that $\gamma_{ij} = \arg \max_{j \in \mathcal{M}} \gamma_{ij}$ and $\gamma_{ij} \geq \gamma_{th}$

where γ_{ij} is the SINR downlink between user i and drone j and γ_{th} is threshold SINR required by the user to successfully have wireless service. Here, the SINR for user i that connects to drone j can be given by:

$$\gamma_{ij} = \frac{\eta P_j d_{ij}^{-\alpha}}{\sum_{u \in \mathcal{I}_{int}} \eta P_u d_u^{-\alpha} + \sigma^2}, \quad (1)$$

$$d_{ij} = \sqrt{(x_i - x_j^D)^2 + (y_i - y_j^D)^2 + h_j^2}, \quad (2)$$

where α is the path loss exponent, σ^2 is the noise power, η is the path loss constant. d_{ij} is the distance between drone-BS j and a given user i . Also, \mathcal{I}_{int} is the set of interfering drone-BSs.

We assume that users have fixed locations and that drones can move to certain locations to service the users. Our goal is to optimally deploy the drones, i.e., calculate optimal locations to provide the wireless service in each of the following scenarios.

A. Maximizing the Number of Covered Users

In the first scenario, our goal is to maximize the number of covered users under limited resources (available drones).

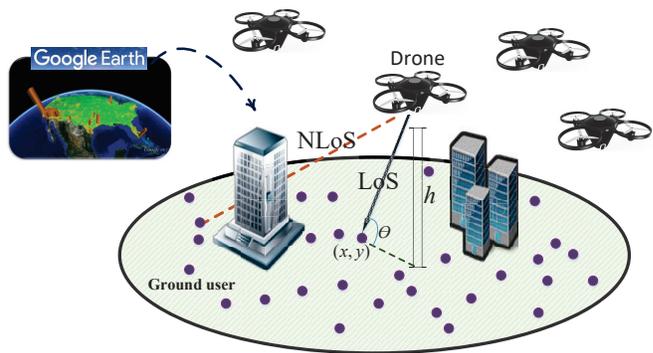


Fig. 1: System model for drones' deployment.

This scenario captures emergency scenarios, e.g., flooding or power outage, or highly unusual wireless service demand, e.g., a fair or a sports event in a stadium. In such cases, the goal of using drones is to provide wireless service to the largest possible number of users. Covering every user in these cases might not be possible due to very high data demand that will require more drones than what is available. Determining the number of drones that can be used in these scenarios depends on the number of available drones and the expected coverage in this geographical area. In emergency cases for example, when more than one geographical area is affected and in need for urgent coverage, drones are to be deployed in these areas according to the percentage of ground users that can be effectively covered with connectivity by the drones.

In this scenario, the number of users is fixed to L and the number of drones is fixed to M . The goal is to find the optimal locations of the drones $\mathbf{v}_j, \forall j \in \mathcal{M}$ to maximize the number of covered users. Let $\mathbb{1}_{ij}$ be an indicator of whether or not user i is connected to drone j such that:

$$\mathbb{1}_{ij} = \begin{cases} 1 & \text{if } j = \arg \max_{j \in \mathcal{M}} \gamma_{ij} \text{ and } \gamma_{ij} \geq \gamma_{th}, \\ 0 & \text{if otherwise.} \end{cases} \quad (3)$$

The problem can then be formulated as:

$$\max_{\mathcal{L}} \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{M}} \mathbb{1}_{ij} \quad (4)$$

$$\text{s. t.} \quad \sum_{j \in \mathcal{M}} \mathbb{1}_{ij} = 1, \forall i \in \mathcal{L}. \quad (5)$$

The constraint in (5) guarantees that every user is connected to only one drone.

B. Full Coverage with a Minimum Number of Drones

In this next scenario, every user needs to be covered using the minimum number of drones. Here, unlike the previous scenario, we do not assume limited resources. This scenario usually occurs in public safety and pre-disaster awareness situations in which every user needs to be informed by a disaster mitigation plan. For example, in pre-disaster evacuation, we need to make sure that every user is aware of the upcoming danger in a timely-manner. This can help improve the community resilience against these type of disasters. Covering every

user (i.e., full coverage) can be achieved by deploying drones in the targeted geographical area. However, as deploying these drones is usually costly, we need to ensure full coverage while minimizing the number of drones, and, hence the cost.

The goal is to calculate the minimum number of drones required to achieve full user coverage to the L available users. This is achieved by calculating the optimal locations of the drones v_j , $\forall j \in \mathcal{M}$ to achieve full coverage of the users. We use the same indicator $\mathbb{1}_{ij}$ as defined in the previous scenario. The problem can then be formulated as:

$$\min_{\mathcal{M}} \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{L}} \mathbb{1}_{ij} \quad (6)$$

$$\text{s. t.} \quad \sum_{j \in \mathcal{M}} \mathbb{1}_{ij} = 1, \forall i \in \mathcal{L}, \quad (7)$$

$$\sum_{i \in \mathcal{L}} \mathbb{1}_{ij} = L. \quad (8)$$

The first constraint ensures that every user is connected to only one drone and the second constraint ensures that all the users are connected to drones.

C. Minimizing Flight Time of Drones in Serving Users

In this third scenario, each user needs to download some data using the wireless service and we are interested in minimizing the hover time (service time) of the drones to satisfy this data load for every user. This scenario captures the case in which the consumed energy is of importance as the drones can provide wireless services for only a limited period of time [15]. One example scenario is the case in which the drones are to be deployed in a geographical area that is far from their source and, thus, the drones will have to consume a significant portion of their energy for traveling to the destination. The remaining amount of energy (that will be used to serve the users) needs to be used in the most effective way possible so as to satisfy the demand of the users.

Each user, among the L users, is assumed to have a load of data given by β_i bits that needs to be satisfied. A drone j can transmit data to a user i with a rate b_{ij} bits/second that depends on γ_{ij} . The time spent by a drone j to serve a user i can then be calculated as:

$$t_{ij} = \frac{\beta_i}{b_{ij}}. \quad (9)$$

The total hover time of a drone j can then be calculated as the summation of the times spent to serve all the users connected to this drone. Let \mathcal{N}_j be the set of all users connected to drone j , $\forall j \in \mathcal{M}$. Then, the hover time for a drone $j \in \mathcal{M}$ will be is given by:

$$t_j = \sum_{i \in \mathcal{N}_j} \frac{\beta_i}{b_{ij}}. \quad (10)$$

The goal in this third scenario is to find the optimal locations of the drones to minimize the overall hover time of all drones

given that the load of each user needs to be satisfied. The problem can be formulated as:

$$\min_{\mathcal{M}} \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{N}_j} \frac{\beta_i}{b_{ij}} \quad (11)$$

$$\text{s. t.} \quad \sum_{j \in \mathcal{M}} \mathbb{1}_{ij} = 1, \forall i \in \mathcal{L} \quad (12)$$

$$\sum_{i \in \mathcal{L}} \mathbb{1}_{ij} = L. \quad (13)$$

The constraints are similar to the previous scenario. In this scenario, when every user is connected to a drone, then every user will be in a set \mathcal{N}_j of a specific drone j such that:

$$\bigcup_{j \in \mathcal{M}} \mathcal{N}_j = \mathcal{L}. \quad (14)$$

Then, the problem formulation of (11) will minimize the overall hover time while ensuring that the total load of users is satisfied.

III. GOOGLE EARTH ENGINE SIMULATOR FOR OBSTACLE LOCATION EXTRACTION

To analyze the aforementioned scenarios, using a ground environment-aware approach, we have developed a drone network simulator using MATLAB that takes as input the locations of buildings. To determine these, we now explore the use of Google Earth Engine, a platform that is suitable for analysis and representation of geospatial data. Earth Engine incorporates multiple datasets and image processing tools. The simplest way to use the Earth Engine is through its built-in JavaScript IDE, which we explore in this work. Python is also supported through an API. The platform is well-suited for our application because of the image processing potential, allowing us to estimate and refine network parameters, and the intuitive interface through which users can supervise the building detection process.

Various building detection algorithms have been developed, with cited precisions ranging from 80-90% [16]–[18]. Accurate algorithms rely on a combination of feature extraction techniques and machine learning. For our application, we circumvent the time and resources needed to train such programs by exploiting the map view imagery supplied by Google. In this view, satellite imagery is simplified, wherein features like buildings are identified in the same color. This greatly facilitates automated building identification, under the assumption that Google's own identification techniques are accurate.

To extract building locations from map view, we use edge detection. This is implemented most readily in Earth Engine through Canny edge detection, a reliable and very common algorithm [19], [20]. Canny detection applies separate filters to detect horizontal, vertical, and diagonal edges, and computes the gradient magnitude. Finally, non-maximum magnitudes are suppressed, thinning the detected edges. In general applications of edge detection, image noise must be accounted for



Fig. 2: Results of building identification imposed over satellite imagery for a region of the Virginia Tech campus.

through the application of Gaussian filters; even then, error is expected. However, the simple, noiseless images provided by map view are ideal candidates for edge detection, and edge detection yields accurate results.

To extract lines from this output, we apply the Hough transform to the Canny image [19]. This step is important to correct imperfections in the Canny output. The Hough transform uses an accumulator to detect the presence of a line, then implements a voting algorithm to identify its parameters. Now, we sample and trace each line, noting changes in direction which correspond to building corners. At this point, we can also manually adjust the locations of any vertex. To examine the accuracy of this process we outlined buildings on map view and overlaid them onto the corresponding satellite imagery, shown in Figure 2.

Evidently, through this method, buildings are approximated fairly well. Over five test cases that we performed, this process correctly outlined about 95% of each building's correct area, and falsely identified an additional 12%. These figures are consistent with the 80-90% accuracy bounds given in the studies cited above. Additionally, we note that this method tends to overestimate building area. This is permissible, and possibly preferable, for UAV simulations, in which drones should not be deployed within a buffer area around buildings, due to the threat of collision. The limitations of the geometric approximation of buildings in this manner include irregular building shapes, specifically ones with rounded sides. Earth Engine only supports polygons; thus, rounded edges must be represented by some number of vertices, adding inherent error. In summary, we have shown that for building location identification, analysis of Google map data is consistent in accuracy with rigorous processing of satellite imagery, but can be performed at reduced computational cost. Thus, while using minimal computational resources, we have identified the locations of buildings, which will be used as inputs into our developed Google Earth-enabled MATLAB simulator so as to analyze the proposed environment-aware wireless drone base station deployment scenarios.

IV. SIMULATION RESULTS

For our simulations, we consider a $200 \text{ m} \times 200 \text{ m}$ area over which users are randomly distributed. Users are assumed to be at ground level, at which $z = 0$. The locations of buildings are known, defined by their vertices at $\{V_1, V_2, \dots, V_N\}$, where each vertex consists of an x and y coordinates. For these simulations, we consider a three-building configuration derived from Figure 2 which is based on a real area from the Virginia Tech campus. As we did not estimate building height during image processing, we model the buildings' z -coordinates as random variables, constrained between 10 and 20 meters, heights appropriate for five-story buildings. Other simulation parameters are listed in Table 1.

To evaluate any arrangement of M drones over N_C candidate points, $\binom{N_C}{M}$ calculations are required. As N_C correlates directly with simulation precision, and hence a large N_C is desirable, the computational complexity can quickly become infeasible. To circumvent this, the following heuristic is implemented. We first discretize the target area into some number N_C of UAV candidate points, where N_C is sufficiently small to enable rapid evaluation. We form the binary power threshold matrix T in which entry (m, n) indicates whether the user at location (x_n, y_n) receives above a given power P_{\min}^t from candidate point m . Note that we do not yet account for interference, noise, or line-of-sight; our current goal is to establish starting points for further optimization. We incrementally place drones at the candidate points is maximized; in other words, at points with the most potential links.

Now, we further discretize the area around each chosen candidate point. Given $\{V_1, V_2, \dots, V_N\}$, we calculate whether a LoS exists by sampling the line segment connecting user i and each candidate point and checking whether any sample point lies within the bounds of a building. If so, we introduce an additional attenuation factor, η , to that potential channel. Finally, we consider interference and noise, and simultaneously solve for the optimal locations of each UAV such that the number of users above a given SINR threshold, γ , is maximized.

Figure 3 shows the percentage of covered users as the SINR threshold needed for connectivity varies (this result corresponds to the deployment scenario in Subsection II-A). Clearly, as the SINR threshold or equivalently the receivers' sensitivity increases, the coverage performance of drones decreases. This due to the fact that satisfying a higher SINR requirement is more challenging thus fewer number of users can be covered by the drones. For instance, when increasing the SINR threshold from 2 dBm to 8 dBm, the number of covered users decreases by 63% in the proposed approach. In Figure 3, we also compare the performance of the proposed deployment approach with a case in which deployment is done based a probabilistic path loss model. In the probabilistic model, a drone can have a LoS link to a ground user with a specific probability, which is given by [21]:

$$P_{\text{LoS},i} = b_1 \left(\frac{180}{\pi} \theta_i - 15 \right)^{b_2}, \quad (15)$$

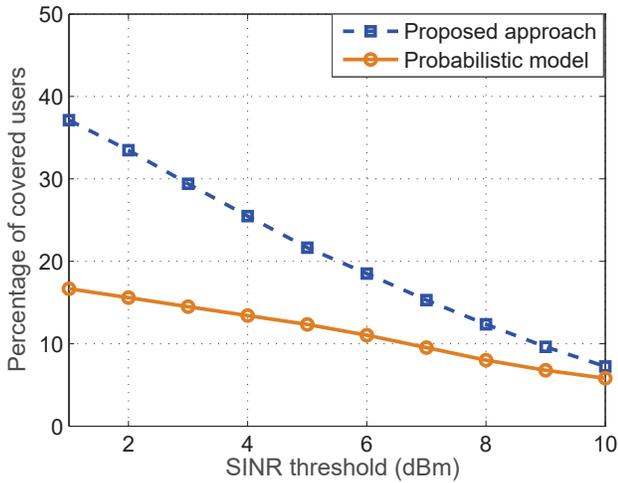


Fig. 3: Percentage of covered users versus SINR threshold.

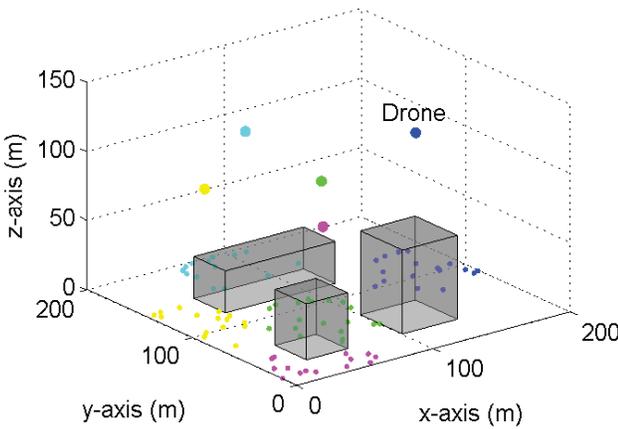


Fig. 4: An illustrative figure for drones' deployment.

where θ_i is the elevation angle (in radians) between the drone i and a user located at (x, y) . Also, b_1 and b_2 are constant values which depend on the environment.

As we can see from Figure 3, our approach outperforms the probabilistic case. In our approach, the locations of buildings are known and deterministic as they are obtained from the Google Earth Engine. In the probabilistic case, however, we do not have a complete information about the buildings. Therefore, by exploiting additional information about the environment, our deployment approach leads to a higher coverage performance than the probabilistic-based deployment. As shown in Figure 3, the number of covered users can be increased by up to a factor of 2 while adopting the proposed environment-aware deployment strategy. As an illustrative example, in Figure 4, we show visual output of drone placement, using known building locations.

Figure 5 shows the impact of the number of drones on the coverage performance for various network sizes (this result corresponds to the deployment scenario in Subsection II-B). Clearly, the coverage performance decreases as the number

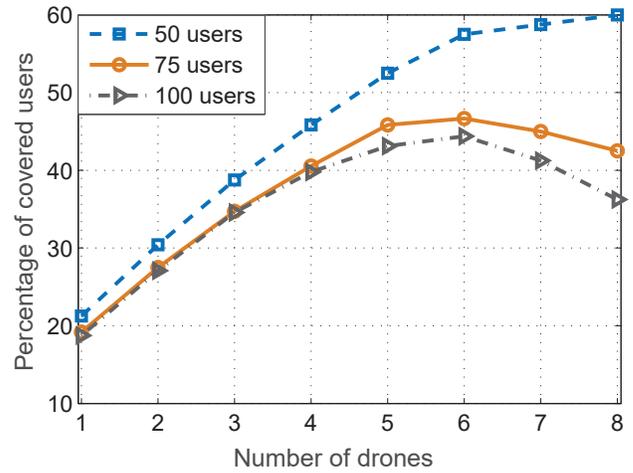


Fig. 5: Percentage of covered users versus number of drones.

ground users increases. For a higher number of users, it will be more likely that drone-users communication links will become blocked by obstacles. Consequently, the communication reliability and, hence, the coverage performance degrades. Figure 5 also shows how the number of covered users varies by changing the number of drones. In this case, there is a tradeoff in deploying more drone base stations for providing wireless connectivity. By increasing the number of drones, the coverage can be improved as the drones are placed closer the ground users. However, while using more drones, the aggregated interference increases which reduces the users' SINR. Therefore, there exists an optimal number of drones for which the coverage is maximized. For instance, as we can see from Figure 5, the optimal number of drones for serving 100 users is 6. This figure allow us to determine the minimum number of drones needed to meet a certain coverage requirement. For example, here, a full coverage for 50 users can be achieved by optimally deploying 8 drones over the considered geographical area.

Figure 6 shows the total flight time of drones needed for completely servicing the users (this result corresponds to the deployment scenario in Subsection II-C). From this figure, we can see that the flight time of drones increases when the number of buildings (i.e., obstacles) increases. With more obstacles in the environment, drone-to-user communications will experience lower SINR due to the blockage and shadowing effects. As a result, the transmission rate will decrease and the drones must fly longer in order to transmit a required amount of data to each user. From Figure 6, it can be seen that the total flight time of drones increases by 45%, in the proposed deployment case, when the number of buildings increases from 1 to 4. Hence, servicing users located in a harsh environment requires longer flight time, more energy consumption, and thus using more capable drones.

In Figure 6, we compare the performance of our proposed environment-aware deployment approach with a random deployment case in which drones are randomly deployed over the geographical area. As we can see from Figure 6, the

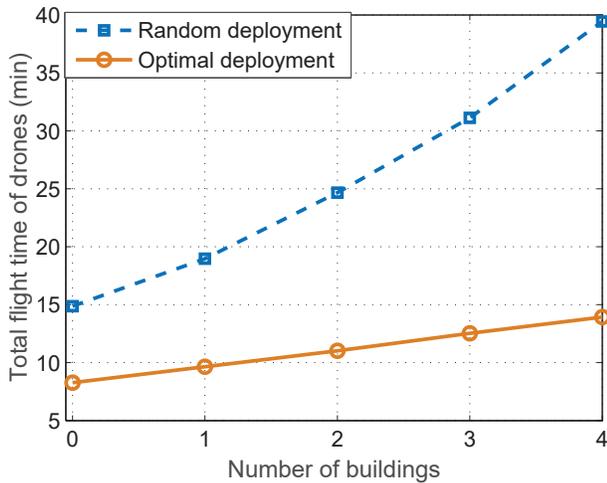


Fig. 6: Total flight time of drones versus number of buildings (i.e., obstacles).

Table I: Simulation parameters.

Parameter	Description	Value
f_c	Carrier frequency	2 GHz
P_i	Drone transmit power	1 W
N_o	Total noise power spectral density	-170 dBm/Hz
N	Number of ground users	200
B	Bandwidth	1 MHz
b_1, b_2	Parameters in probabilistic channel model	0.36, 0.21
β	Load per ground user	10 Mb

proposed optimal deployment can yield up to a 65% flight time reduction compared to the random deployment case. Therefore, the proposed approach enhances energy-efficiency of the considered drone-enabled wireless network.

V. CONCLUSION

In this paper, we have investigated the problem of environment-aware deployment of drone base stations that provide wireless connectivity to ground users. To this end, first, we have developed a drone network simulator that uses the Google Earth Engine in order to extract key information about buildings in the considered geographical area. Then, we have studied the optimal deployment of drones in three practical scenarios. In the first scenario, we have determined the optimal locations of drones such that the number of covered ground users is maximized. In the second scenario, we have minimized the number of drones needed to ensure a full coverage for all users. Finally, we have minimized the flight time of drones required to completely service the users by satisfying their load requirements. Our results have shown that the proposed deployment framework significantly enhances the drone wireless system performance in terms of coverage and energy efficiency. Moreover, our simulation results have demonstrated existence of an optimal number of drones for which the wireless coverage is maximized.

REFERENCES

- [1] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *available online: arxiv.org/abs/1803.00680*, 2018.
- [2] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, Aug. 2017.
- [3] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for uav-enabled multiple access," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, Singapore, Dec. 2017.
- [4] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: Multi-tier drone-cells," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–55, Nov. 2016.
- [5] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G with UAVs: Foundations of a 3D wireless cellular network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 357–372, Jan. 2019.
- [6] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, June 2016.
- [7] A. R. Khan, S. M. Bilal, and M. Othman, "A performance comparison of open source network simulators for wireless networks," in *Proc. of IEEE International Conference on Control System, Computing and Engineering*, Penang, Malaysia, Nov. 2012.
- [8] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, "Comparative study of wireless network simulators," in *Proc. of International Conference on Networking*, Cancun, Mexico, Apr. 2008.
- [9] S. Kang, M. Aldwairi, and K.-I. Kim, "A survey on network simulators in three-dimensional wireless ad hoc and sensor networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 10, p. 1550147716664740, 2016.
- [10] B. Newton, J. Aikat, and K. Jeffay, "Simulating large-scale airborne networks with ns-3," in *Proc. of the ACM Workshop on ns-3*, Barcelona, Spain, May 2015.
- [11] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1647–1650, Aug. 2016.
- [12] E. Kalantari, H. Yanikomeroglu, and A. Yongacoglu, "On the number and 3D placement of drone base stations in wireless cellular networks," in *Proc. of IEEE Vehicular Technology Conference*, 2016.
- [13] M. M. Azari, F. Rosas, K. C. Chen, and S. Pollin, "Joint sum-rate and power gain analysis of an aerial base station," in *Proc. of IEEE GLOBECOM Workshops*, Dec. 2016.
- [14] O. Esrafilian, R. Gangula, and D. Gesbert, "Learning to communicate in UAV-aided wireless networks: Map-based approaches," *IEEE Internet of Things Journal*, *Early access*, 2018.
- [15] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8052–8066, Dec. 2017.
- [16] A. Zhang, X. Liu, A. Gros, and T. Tietcke, "Building detection from satellite images on a global scale," *available online: arxiv.org/abs/1707.08952*, 2017.
- [17] M. Cote and P. Saeedi, "Automatic rooftop extraction in nadir aerial imagery of suburban regions using corners and variational level set evolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 313–328, Jan. 2013.
- [18] J. P. Cohen, W. Ding, C. Kuhlman, A. Chen, and L. Di, "Rapid building detection using machine learning," *Applied Intelligence*, vol. 45, no. 2, pp. 443–457, 2016.
- [19] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [20] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [21] A. Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Austin, TX, USA, Dec. 2014.