Document downloaded from:

http://hdl.handle.net/10251/131278

This paper must be cited as:

Prado-Alvarez, D.; Inca-Sanchez, SA.; Martín-Sacristán, D.; Monserrat Del Río, JF. (2019). Comparison of Optimization Methods for Aerial Base Station Placement with Users Mobility. IEEE. 485-489. https://doi.org/10.1109/EuCNC.2019.8802053

The final publication is available at

https://doi.org/10.1109/EuCNC.2019.8802053

Copyright IEEE

Additional Information

Comparison of Optimization Methods for Aerial Base Station Placement with Users Mobility

Danaisy Prado, Saul Inca, David Martín-Sacristán and Jose F. Monserrat
Universitat Politècnica de València, iTEAM Research Institute, Camino de Vera s/n, 46022 Valencia, Spain
Email: dapraal@iteam.upv.es

***Abstract*—Aerial base stations have been recently considered in the deployment of wireless networks. Finding the optimal position for one or multiple aerial base stations is a complex problem tackled by several works. However, just a few works consider the mobility of the users which makes necessary an online optimization to follow the changes in the scenario where the optimization is performed. This paper deals with the online optimization of an aerial base station placement considering different types of users mobility and three algorithms: a Q-learning technique, a Gradient-based solution and a Greedy-search solution. Our objective is to minimize in an urban environment the path loss of the user at street level with the highest path loss. Simulation results show that the performance of the three methods is similar when a high number of users move randomly and uniformly around the scenario under test. Nevertheless, in some situations when the number of users is reduced or when the users move together in a similar direction, both Gradient and Greedy algorithms present a significantly better performance than the Q-learning method.**

I. Introduction

Aerial base stations have been recently considered in the deployment of wireless networks with many purposes. For example, in emergency responses and public safety situations such as earthquakes, hurricanes, fires or in military operations where terrestrial communications networks are damaged or not fully operational, a fast deployment of communication systems is crucial. Seizing the versatility of Unmanned Aerial Vehicles (UAVs), instant networks based on them are often performed for these cases since the UAV networks can overfly the area very quickly avoiding physical obstacles [1].

Finding the optimal position for one or multiple aerial base stations is a complex problem tackled by several works. Some of those works deal with that problem analytically or use heuristic algorithms such as particle swarm optimization [2] considering basically a static scenario and hence the optimization for a single snapshot of an environment. Just a few works consider the mobility of the users which makes necessary an online optimization, i.e. an optimization with incomplete knowledge of the future, able to follow the changes in the scenario where the optimization is performed. In both [3] and [4], a reinforcement learning technique, Q-learning, is used to perform the online optimization and maximize the sum of the rates of the users. In [3], it is optimized the 3D position of an aerial base station which complements a terrestrial network, whereas, in [4], the focus is on the deployment of multiple aerial base stations in an urban area.

However, in those works the Q-learning method is not compared with any other non-static positioning of the aerial base stations. Additionally, those works consider the maximization of a global metric such as the sum-rate, but in some cases it could be useful to focus on a worst-user metric. Consider, for example, an emergency scenario where the communication networks have been destroyed and a single aerial base station is temporarily used to provide direct communications among the users or even external communications. In that case, it may be important to maximize the rate of the user with worst channel, and the maximization of the sum of the rates could unfortunately lead to that user experiencing even null rates. To avoid such problem, we focus on the minimization of the maximum path loss experienced by all the users in certain scenario. Given that this maximum path loss is a worst-user metric and not a global metric, it is more likely to find high variations due to the users movement. These variations may affect the ability of different methods to perform an online optimization of the aerial base station placement, which results in the need to compare different online optimization methods.

This paper deals with the online optimization of a single aerial base station placement in an urban scenario to minimize the maximum path loss between a set of users located in the street level and the aerial base station considering different types of users mobility and three optimization algorithms: a Q-learning technique, a Gradient-based solution and a Greedy-search solution.

The remainder of the paper is as follows. Section II clarifies the optimization problem addressed. Then, Section III presents the three methods compared in this paper for online optimization of aerial base station placement. Section IV presents the simulation environment employed to obtain the results shown in Section V. Finally, the conclusions of the assessment are discussed in Section VI.

II. Problem statement

We consider a set of N users where the i-th user has at a certain time a position (x_i, y_i, z_i), where z_i is 1.5 m for all the users. The position of the aerial base station is (x_{BS}, y_{BS}, z_{BS}). For the sake of simplicity, we consider a fixed height for the aerial base station, therefore z_{BS} is fixed and we will exclude the coordinate z from the optimization problem. The path loss for the channel between the i-th user and the aerial base station is l_i, which depends on the positions of both channel ends.

Our optimization problem tries to find the base station position (x^*_{BS}, y^*_{BS}) which minimizes the maximum path loss for the channels between the users in the scenario and the aerial base station. We denote that maximum path loss as L and we will refer to it as the worst-user path loss.

The optimization problem can be formulated as:

$$\begin{aligned} \min_{x^*_{BS}, y^*_{BS}} \quad & L := \max(\{l_1, ..., l_N\}) \\ \text{subject to} \quad & x_{min} \leq x_{BS} \leq x_{max} \\ & y_{min} \leq y_{BS} \leq y_{max} \end{aligned}$$

Due to the users movement, the path loss between the users and the aerial base station changes continuously, and hence the worst-user path loss. To denote such variation we will use L_t to refer to the worst-user path loss at time t. Additionally, we will use $L(x, y)$ or $L(P)$ to refer to the worst-user path loss given a specific location of the aerial base station. The user movement makes necessary a continuous optimization of the aerial base station placement which will be the aim of the online optimization methods presented in the following section.

III. Proposed solutions

In this section we present the three online optimization solutions evaluated in this assessment.

A. Q-learning solution

Reinforcement Learning (RL) is a type of machine learning technique focused on how to take the best actions at specific states of an environment to maximize certain reward. RL techniques explore the environment in a first phase to learn a knowledge used in a subsequent exploitation phase [4].

Q-learning is a reinforcement learning method where the learner builds incrementally a Q-function which attempts to estimate the discounted future rewards for taking actions from given states [5]. In a Q-learning algorithm, the agent considers a class of states $S = \{s_1, s_2, ..., s_n\}$, a class of actions $A = \{a_1, a_2, ..., a_m\}$, and a knowledge matrix Q. In each state, the learning agent performs an action a_t which triggers a state transition [3]. Then, the agent calculates the reward in the new state, and the matrix Q is updated. In this paper, we use Q-learning to find the optimal position of the aerial base station which minimizes the path loss of the user with highest path loss. In the proposed solution, the area of interest is selected and divided into a grid where the vertices of the grid are used as states of the Q-learning algorithm. Therefore, each state corresponds to a position. The agent, an UAV, can take, in general, in each state a set of nine actions comprising the eight possible movements to neighbour vertices (move left, move right, move forward, move backward and move in the four diagonal directions) together with the action consisting in staying in the same position. The set of actions is reduced in states located in the borders of the area of interest. Since it is intended to minimize the path loss, the reward is defined as the inverse of the worst-user path loss measured which allows to get the greater rewards for the lower path loss.

Our implementation is detailed in Algorithm 1. The initial state is randomly selected and the matrix Q, of size $n \times m$, where n is the number of states and m the number of actions, is initialized to zero. The algorithm is implemented in two phases known as exploration and exploitation. During the exploration, the learner takes actions to learn the reward that can be expected from each action in each state. With this aim, several iterations are carried out whose exact value, $I_{exploration}$, depends on the number of states considered since the higher the number of states, the higher the number of possible transitions whose reward must be learned. In our case, the exploration is made according to an $\epsilon - greedy$ strategy. An ϵ value, which is decreasing with the number of iterations, is used to decide if the action taken in each state is randomly chosen or selected to maximize the expected reward. With each iteration the brain of the agent, represented by the Q matrix, is enhanced thanks to the update of Q shown in line 14. The variable α is the learning rate that indicates to which extent new information overrides old information. The discount rate is represented by $\lambda_{q-learning}$ which determines how much future rewards are worth, compared to the value of immediate rewards. Both α and $\lambda_{q-learning}$ take values between 0 and 1 (0.9 in our simulations) [3] [4]. The exploration phase progresses until matrix Q is stable, which means our agent has learned enough, so exploitation phase can start. During the exploitation phase the agent takes in each state the optimum action according to matrix Q [3]. In our case, in the exploitation phase the Q matrix is also updated to enable a continuous learning that is required to adapt the decisions to the continuous movement of the users.

Algorithm 1 Q-learning solution

1: Create n states
2: Initialize Q as a zero matrix of size $n \times m$
3: Randomly chose initial state s_0
4: **for** $t = 1$ to $I_{exploration}$ **do**
5: Draw a random number u_t
6: Calculate ϵ_t according to a decreasing function
7: **if** $u_t < \epsilon_t$ **then**
8: Randomly chose an action a_t
9: **else**
10: Find best action: $a_t = \arg\max_{a \in A} Q(s_t, a)$
11: **end if**
12: Perform the action a_t
13: Compute reward $r_{t+1} = \frac{1}{L_t}$
14: Update Q: $\mathrm{Q}(s_t, a_t) \leftarrow \mathrm{Q}(s_t, a_t) + \alpha(r_{t+1} + \lambda_{q-learning} \max_{\{a\}} \mathrm{Q}(s_{t+1}, a) - \mathrm{Q}(s_t, a_t))$
15: Current state $s_t \leftarrow$ Next state s_{t+1}
16: **end for**
17: **loop**
18: Find best action: $a_t = \arg\max_{a \in A} Q(s_t, a)$
19: Perform the action a_t
20: Compute reward $r_{t+1} = \frac{1}{L_t}$
21: Update Q: $\mathrm{Q}(s_t, a_t) \leftarrow \mathrm{Q}(s_t, a_t) + \alpha(r_{t+1} + \lambda_{q-learning} \max_{\{a\}} \mathrm{Q}(s_{t+1}, a) - \mathrm{Q}(s_t, a_t))$
22: Current state $s_t \leftarrow$ Next state s_{t+1}
23: **end loop**

B. Gradient-based solution

Online optimization can also be achieved thanks to a gradient-based solution. Gradient-based optimization is an iterative methodology in which steps are taken in the solution space proportionally to the gradient of the function to be optimized. If the gradient of the function is not exactly known, as in our real-world optimization example, it can still be approximated through finite differences. In this case, after a movement in one space dimension, the partial derivative over that dimension can be approximated dividing the increment of the function after that movement by the shift performed in that dimension. In our case, we consider as that shift the one decided by the gradient algorithm in the previous iteration except in the first calculation in which the shift is a fixed value. See a detailed description in Algorithm 2 where x and y are the coordinates of the UAV and $L(x, y)$ is the path loss of the user with highest path loss when the UAV is in the position (x, y). The variable $\lambda_{gradient}$ is the proportionality constant between the steps of the algorithm and the gradient. In our simulations, the optimum value for $\lambda_{gradient}$ was 10.

Algorithm 2 Gradient-based solution

Initialize x shift: $\Delta x = 1$
Initialize y shift: $\Delta y = 1$
$\lambda_{gradient} = 10$
$i = 0$
$x_0 = x$
$y_0 = y$
Measure $L(x_i, y_i)$
loop
  $x_{i+1} = x_i + \Delta x$
  $y_{i+1} = y_i + \Delta y$
  Move to (x_{i+1}, y_i)
  Measure $L(x_{i+1}, y_i)$
  Move to (x_{i+1}, y_{i+1})
  Measure $L(x_{i+1}, y_{i+1})$
  $\frac{\partial L}{\partial x} = \frac{L(x_{i+1}, y_i) - L(x_i, y_i)}{\Delta x}$
  $\frac{\partial L}{\partial y} = \frac{L(x_{i+1}, y_{i+1}) - L(x_{i+1}, y_i)}{\Delta y}$
  Set new x shift: $\Delta x = -\lambda_{gradient} \frac{\partial L}{\partial x}$
  Set new y shift: $\Delta y = -\lambda_{gradient} \frac{\partial L}{\partial y}$
  $i = i + 1$
end loop

C. Greedy-search solution

As other greedy-search algorithms, our proposal makes locally optimal choices at each stage targeting a global optimum. To be more specific, in each step of the algorithm, the UAV moves into a set of neighbour positions around the initial position and, for each neighbour position, the UAV checks if the value of the function to be optimized, i.e. the path loss of the user with highest path loss, is lower than in the initial position. If the condition is met for any neighbour position, the UAV stops the current iteration and sets that neighbour position as the initial position of the following iteration. If the condition is not met by any neighbour, the UAV returns to the initial position and starts a new iteration. A detailed description of the algorithm is shown in Algorithm 3. Note that we have used 8 neighbour positions (P_1 to P_8) around the initial position (P_0) for each iteration. This set of neighbour positions is the result of the combination of three possible shifts in each axis: $\{-\Delta x, 0, +\Delta x\}$ in x axis and $\{-\Delta y, 0, +\Delta y\}$ in y axis, where the values of Δx and Δy are 1 m in our tests. Again, $L(P_n)$ is the path loss of the user with highest path loss when the UAV is in the position P_n.

IV. Simulation Environment

We have developed a simulated scenario in Unity, as in [6], which is a multi-platform game engine to develop realistic virtual environments in a cost-efficient manner. The following sections describe the simulated environment.

Algorithm 3 Greedy-search solution

Initialize x shift: $\Delta x = 1$
Initialize y shift: $\Delta y = 1$
loop
  $P_0 = (x, y)$
  $P_1 = (x, y + \Delta y)$
  $P_2 = (x + \Delta x, y + \Delta y)$
  $P_3 = (x + \Delta x, y)$
  $P_4 = (x + \Delta x, y - \Delta y)$
  $P_5 = (x, y - \Delta y)$
  $P_6 = (x - \Delta x, y - \Delta y)$
  $P_7 = (x - \Delta x, y)$
  $P_8 = (x - \Delta x, y + \Delta y)$
  Measure $L(P_0)$
  for $i = 1$ to 8 **do**
    Move to P_i
    Measure $L(P_i)$
    if $L(P_i) < L(P_0)$ **then**
      Reinitialize loop
    end if
  end for
  Move to P_0
end loop

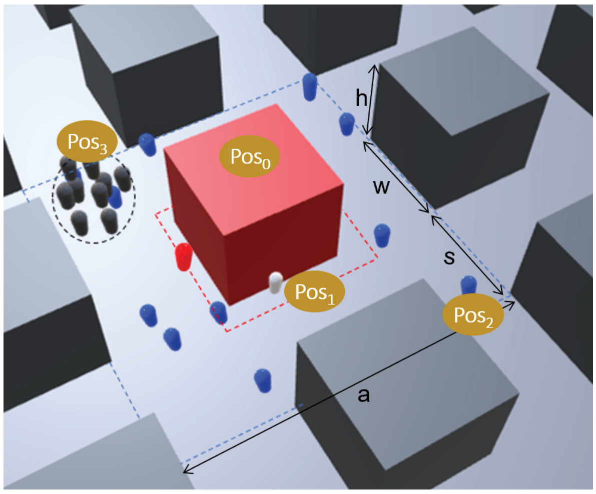

Fig. 1. Urban scenario developed in Unity with different user mobility patterns and aerial base station static positions

A. Urban environment

In this paper, an urban scenario has been deployed following the urban high-rise environment in [7]. The scenario comprises evenly space buildings as shown in Fig. 1. The height of the buildings, h, is 18 meters, which is in the range for urban scenarios. The width, w, and the separation between them, s, are 20 meters. The users are dropped within the area of a $\times$ a m^2 that surrounds the red building, where a = w + 2s.

B. Propagation model

The Third Generation Partnership Project (3GPP) channel model in [8] describes a large-scale propagation path loss at frequency below 6 GHz for several deployment scenarios, focused on outdoor and indoor environments. In this assessment we use the models for UMa macro-cell scenario in which the base stations are mounted above rooftop levels of surrounding buildings, similar to the aerial base stations considered in this work. In our simulations, the centre frequency is 2 GHz and the heights of the base station and the user terminal are 30 m and 1.5 m, respectively. In 3GPP path loss model [8], a probability of being in Line Of Sight (LOS) is given [9]. In our tests, the probability function has been replaced by a single ray-trace technique, very simple in the Unity environment, which consist of throwing a ray from the base station to the user and checking if the ray is blocked by a building, user or other element and also check if the Fresnel zone is blocked or not.

C. Drone and users mobility

We consider a maximum speed for the UAVs of 45 km/h [10]. The exact UAV mobility depends on the placement optimization method used. For the sake of comparison, we also consider an static placement of the drone. In Fig. 1 three different points for drone static placement are depicted with a yellow spot and denoted by Pos_0, Pos_1, Pos_2 and Pos_3.

Fig. 1 also illustrates the user mobility patterns considered in this assessment. The user in red circles the building following the red dotted line route always with the same direction. The user in white moves randomly around the central building changing its direction from time to time. These two movement patterns could be followed, e.g., by a group of people on a patrol around a building or by an urban search and rescue team looking for survivors in a collapsed building. The set of users in blue perform a random movement over the whole area being uniformly distributed, and can be seen as a general pattern for emergency response teams, while the users in gray start moving like the blue ones and meet at the zone delimited by a dotted line in gray, which may represent an evacuation operation. In all cases, users move at 5 km/h.

V. PERFORMANCE EVALUATION

In this section we assess the three online optimization methods presented in four possible study cases where the number and mobility of users are different.

A. Ten users moving randomly and uniformly distributed

A simulation has been carried out with ten users distributed and moving randomly within the square area delimited by a blue dotted line in Fig. 1 (see the blue users). Fig. 2 shows the Cumulative Distribution Function (CDF) for the path loss of the worst user. It shows almost identical results for the static placement in Pos_0, Q-learning, Gradient, and Greedy algorithms because the optimization algorithms place the UAV close to Pos_0 which is the optimum static position in this case. On the other hand, a worse path loss is obtained in Pos_1 and Pos_2 where Non Line Of Sight (NLOS) is more frequent for the channels between the users and the UAV than in Pos_0.

B. One user circling around the central building

In this case we consider a single user circling around the central building (see the red user in Fig. 1). It may also represent a group of users performing the same movement together. Fig. 3 shows the CDF for the path loss of that user. Concerning the static placement, in Pos_0 the LOS is completely obstructed by the central building. In Pos_1, the user is in LOS less than the 25% of the time while in Pos_2 there is line of sight a 50% of the time. Regarding the Q-learning, Gradient and Greedy curves, the three algorithms are able to follow the user in its movement and keep the LOS, however the Greedy and the Gradient behave better and are slightly shifted towards lower path loss values compared to Q-learning as these algorithms have more spatial granularity in the placement of the UAV than Q-learning.

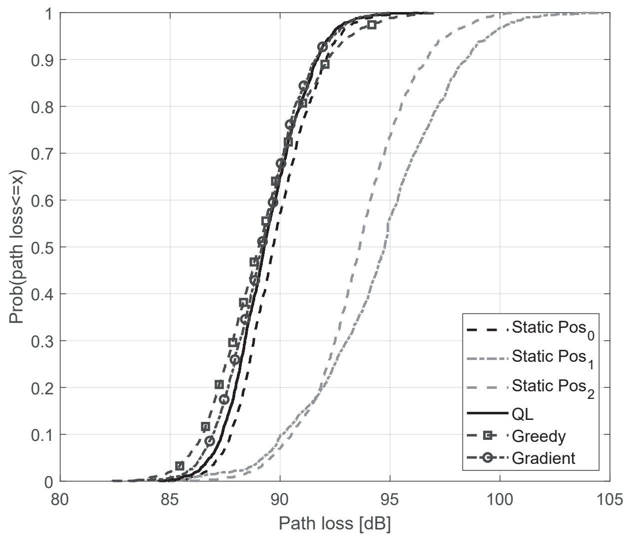

Fig. 2. Worst-user path loss CDF in the case of 10 users moving randomly and uniformly

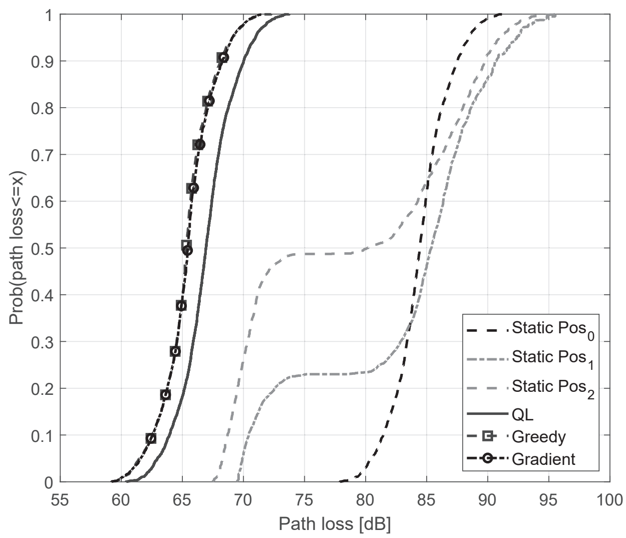

Fig. 3. Path loss CDF for one user circling around the central building

C. One user moving randomly around the central building

If a user moves close to the central building but changes its direction from time to time (see white user in Fig. 1) we obtain the results shown in Fig. 4. For the static positions, the

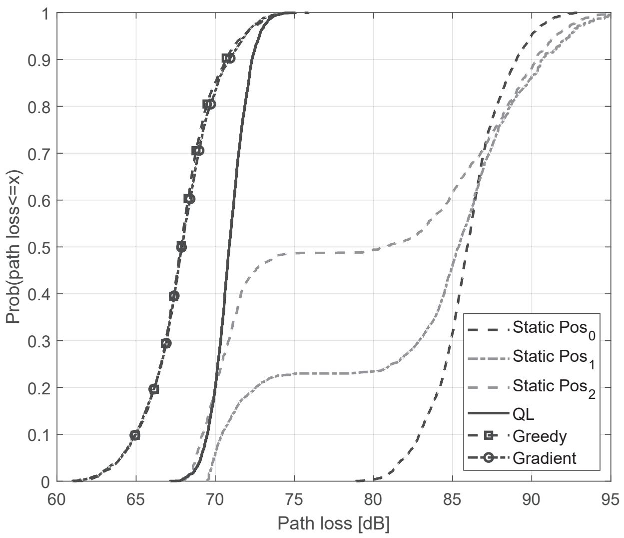

Fig. 4. Path loss CDF in the case of one user moving randomly around the central building

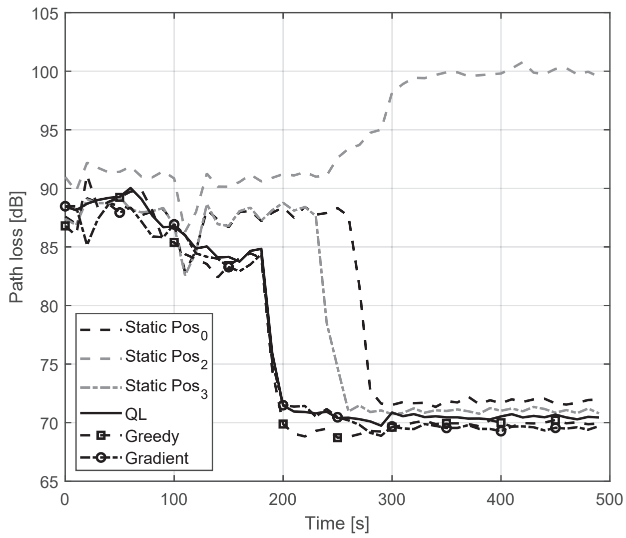

Fig. 5. Temporal evolution of the worst-user path loss in the case of 10 users meeting at one point

results look similar to those of the previous case. However, a worst performance can be appreciated for Gradient and Greedy algorithms, and especially for Q-learning. This fact suggests that it is harder for this algorithm to follow the user when the user movement direction changes suddenly.

D. Ten users meeting at one point

In this case, the simulation starts with 10 users allocated where the blue ones are in Fig.1. As the simulation progresses all the users move to the same zone delimited by a dotted line in gray where they stand still. Fig. 5 shows a temporal evolution of the path loss of the worst user. In the static placement case, in Pos_0 and Pos_3, the path loss values improve at some point in time in which all the users are in LOS. See that the meeting point is in LOS with regard to those points in Fig. 1. However, in Pos_2, all the users are in NLOS at some point in time which is reflected in the high path loss values measured in that position. Thus, a static placement can be very detrimental in this case. The Q-learning, Gradient and Greedy algorithms allow the adaptation of the position of the aerial base station and reduce the path loss levels similarly at least 60 seconds before compared to the static positions.

VI. Conclusion

In this paper we have compared three methods of online optimization for aerial base station placement in an urban environment with users mobility. Those methods are based on Q-learning, Gradient calculation and a Greedy-search. Simulation results show that the performance of the three methods is similar when a high number of users move randomly and uniformly around the scenario under test. Nevertheless, in some situations when the number of users is reduced or when the users move together in a similar direction, both Gradient and Greedy algorithms present a significantly better performance than the Q-learning method. Future work will include a larger scenario and the 3D optimization of the aerial base station placement.

Acknowledgment

The work of Danaisy Prado was supported by the H2020 Marie Curie Program, with Project Grant No. 766231 WAVE-COMBE - ITN - 2017.

References

[1] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: leveraging UAVs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, Jan. 2017.

[2] E. Kalantari, H. Yanikomeroglu, and A. Yongacoglu, "On the number and 3D placement of drone base stations in wireless cellular networks," in *IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Montreal, QC, Canada, Sep. 2016, pp. 1–6.

[3] R. Ghanavi, E. Kalantari, M. Sabbaghian, H. Yanikomeroglu, and A. Yongacoglu, "Efficient 3D aerial base station placement considering users mobility by reinforcement learning," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain, Apr. 2018, pp. 1–6.

[4] P. V. Klaine, J. P. Nadas, R. D. Souza, and M. A. Imran, "Distributed drone base station positioning for emergency cellular networks using reinforcement learning," *Cognitive computation*, vol. 10, no. 5, pp. 790–804, May 2018.

[5] D. Pandey and P. Pandey, "Approximate Q-learning: An introduction," in *Second International Conference on Machine Learning and Computing*, Bangalore, India, May 2010, pp. 317–320.

[6] D. Garcia-Roger, D. Martín-Sacristán, S. Roger, J. F. Monserrat, A. Kousaridas, P. Spapis, S. Ayaz, and C. Zhou, "5G multi-antenna V2V channel modeling with a 3D game engine," in *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Barcelona, Spain, Apr. 2018, pp. 284–289.

[7] ITU-T, "Propagation data and prediction methods for the planning of short-range outdoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz," International Telecommunication Union, Recommendation P.1411-9, Jun. 2017.

[8] 3GPP TSG RAN, "Study on channel model for frequency spectrum above 6 GHz," 3GPP, TR 38.900 V14.3.1, Jul. 2017.

[9] ITU-T, "Propagation data and prediction methods required for the design of terrestrial broadband millimetric radio access systems operating in a frequency range of about 20-50 GHz," International Telecommunication Union, Recommendation P.1410-2, Nov. 2003.

[10] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Communications Surveys Tutorials*, 2019, in press.