

Multi-Objective Deep Reinforcement Learning for 5G Base Station Placement to Support Localisation for Future Sustainable Traffic

Ahmed Al-Tahmeesschi¹, Jukka Talvitie², Miguel López-Benítez^{3,4}, Hamed Ahmadi¹, and Laura Ruotsalainen⁵

¹School of Physics Engineering and Technology, University of York, United Kingdom

²Unit of Electrical Engineering, Tampere University, Tampere, Finland

³Dept. of Electrical Engineering and Electronics, University of Liverpool, United Kingdom

⁴ARIES Research Centre, Antonio de Nebrija University, Spain

⁵Dept. of Computer Science, University of Helsinki, Helsinki, Finland

Abstract—Millimeter-wave (mmWave) is a key enabler for next-generation transportation systems. However, in an urban city scenario, mmWave is highly susceptible to blockages and shadowing. Therefore, base station (BS) placement is a crucial task in the infrastructure design where coverage requirements need to be met while simultaneously supporting localisation. This work assumes a pre-deployed BS and another BS is required to be added to support both localisation accuracy and coverage rate in an urban city scenario. To solve this complex multi-objective optimisation problem, we utilise deep reinforcement learning (DRL). Concretely, this work proposes: 1) a three-layered grid for state representation as the input of the DRL, which enables it to adapt to the changes in the wireless environment represented by changing the position of the pre-deployed BS, and 2) the design of a suitable reward function for the DRL agent to solve the multi-objective problem. Numerical analysis shows that the proposed deep Q-network (DQN) model can learn/adapt from the complex radio environment represented by the terrain map and provides the same/similar solution to the exhaustive search, which is used as a benchmark. In addition, we show that an exclusive optimisation of coverage rate does not result in improved localisation accuracy, and thus there is a trade-off between the two solutions.

Index Terms—5G, deep Q-learning, deep reinforcement learning, multi-objective optimisation

I. INTRODUCTION

At present, road transport contributes a significant amount to the total carbon dioxide (CO₂) emissions in the EU [1]. Thus, cities are looking for practical strategies to make their transport systems more intelligent, efficient and sustainable. One promising solution is in the form of connected automated vehicles (AV). Next-generation transportation systems represented by AVs will require Vehicle-to-Infrastructure (V2I) wireless connectivity [2]. Such connection should be able to satisfy the required high-data rates, low latency and decimeter localisation accuracy [3].

5G networks have already adopted millimetre wave (mmWave) along with massive multiple-input-multiple-output (MIMO) technologies to provide extremely high data rates, low latency and localisation. However, due to unfavourable propagation conditions at high frequencies, mmWave signals experience higher path-loss and are more susceptible to build-

ing blockages than sub-6 GHz bands in urban scenarios [4, 5]. Therefore, careful planning of the base station (BS) locations is essential to reduce infrastructure costs while maintaining the quality of service and localisation accuracy [6].

Reinforcement learning (RL) is a promising technique that can be employed to address this problem. RL deals with sequential decision-making problems. The goal of a sequential decision-making problem is to select actions to maximize long-term rewards [7]. RL and the deep RL (DRL) variant have been used in the literature to optimise various wireless communications systems, for example, relay nodes selection [8] and dynamic spectrum access channel selection [9]. More details on RL application to wireless communications can be found in [10]. In this work, we propose the utilisation of DRL to jointly optimise the coverage rate and localisation accuracy.

The problem of BS location optimisation has already been addressed by several studies [11–13] utilising genetic algorithms or computational geometry combined with optimisation tools. In addition, DRL algorithms have been mainly used for aerial BS placement to operate alone or to support terrestrial network infrastructure to improve users coverage and throughput. For instance, a single deep Q-network (DQN) agent is used to control either a single aerial BS [14, 15] or multiple aerial BSs [16]. A multi-agent RL (MARL) approach is utilised to control multiple aerial BSs [17]. In the aforementioned works, the DRL has been mainly used for aerial BS location placement, while our work considers street level base stations. In addition, our work assumes a pre-deployed BS in the service region, and the proposed algorithm is capable to adapt for changes in the pre-deployed BS location. The contributions of this work are outlined as follows:

- 1) We investigate the BS placement with the objective of jointly optimising the coverage rate and localisation accuracy, particularly in the presence of a pre-deployed BS. This addresses the challenge of achieving both coverage requirements and accurate localisation in urban city scenarios using mmWave technology.
- 2) We propose a solution based on DQN to tackle the multi-objective problem. The DQN framework incorporates a

novel state representation approach, using a three-layered grid, which enhances the adaptation to the dynamic radio environment. Additionally, we design a suitable reward function to guide the DQN agent towards finding solutions that balance between coverage rate and localisation accuracy.

- 3) We demonstrate the effectiveness the DQN framework in adapting to changes in the radio environment, as represented by the repositioning of the pre-deployed BS. The DQN model, utilising the proposed state representation approach, showcases the capability to learn and adjust in complex radio environments.

II. 5G NEW RADIO NETWORK MODEL

In this section, the 5G radio network model is described. In order to generate a realistic wireless simulation environment, the mmWave signals are generated using a ray-tracing-based approach, as recommended by 3GPP [18]. Furthermore, as our city model we select the Madrid grid, developed by the METIS project [19], to represent a generic European city layout. The realizations of the ray-tracing-based radio channel are evaluated by using Wireless InSite@software [20]. A similar environment has been considered in the literature [21, 22].

For this work, a specific segment of the Madrid grid is selected with BSs operating at 28 GHz. The BSs height are set to 9 m and each BS includes 4 sectors, where each sector includes a uniform linear array with 32 half-wavelength-separated patch antenna elements. The azimuth orientations of the sectors are set to 45°, 135°, 225° and 315°. The utilised beamforming technique is implemented following the phased-array principle with a total of 64 beams per BS. The transmit power is set to 10 dBm. Fig. 1 shows the considered Madrid grid segment along with the candidate BSs locations. In this work, the received signal strengths (RSSs) are utilised for the estimation of the area coverage rate and localisation accuracy. In practice, the RSS measurements are obtained by the UE based on 3GPP-specified synchronisation signal blocks (SSBs) transmitted by each BS over the 64 beams [23]. One of the clear benefits of the considered approach is that the SSBs are continuously available in all 5G NR networks as part of standard network operation, and thus there is no need for dedicated positioning reference signals whose availability can considerably vary in practical deployments. Moreover, RSS measurements are preferred, as they are available in the user device during both the connected mode and the idle mode as part of underlying mobility management procedures.

III. PROBLEM FORMULATION

Knowing the system setup, we present the BS placement strategy to jointly optimise the localisation accuracy and the coverage rate. After that we also discuss traditional exhaustive search algorithms.

A. Optimisation problem

We have a multi-objective optimisation problem where we need to find the optimal location of the BSs to minimise the average error in localisation and maximise the coverage rate. The optimisation problem can be formulated as:

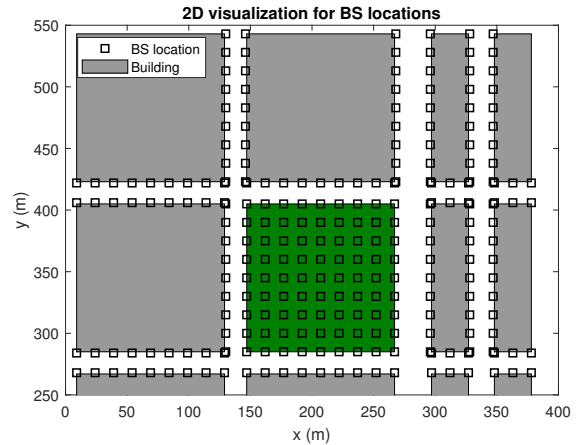


Fig. 1. Madrid grid segment with candidate BS locations.

$$\begin{aligned} \max\{f_1(x, y)\} \wedge \min\{f_2(x, y)\} \\ \text{s.t. } x \in X \\ y \in Y \end{aligned} \quad (1)$$

where X and Y represent the sets of potential coordinates for the BS, $f_1(x, y)$ is the average coverage rate (in percent) of the area and $f_2(x, y)$ is the average localisation error (in meters). The average coverage rate $f_1(x, y)$ is given as

$$f_1(x, y) = \frac{1}{N} \sum_{n=1}^N C_n, \quad (2)$$

where N is the number of positions to be covered and C_n is the coverage rate and is given as

$$C_n = \begin{cases} 1 & \text{if } \text{RSS} \geq \delta, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where RSS is the received signal strength from one of the BS beams and δ is the threshold for the minimum power required for a correct signal reception. The average localisation error $f_2(x, y)$ is computed as

$$f_2(x, y) = \frac{1}{N} \sum_{n=1}^N z_n, \quad (4)$$

where z_n is the Euclidean distance between the estimated user equipment (UE) horizontal plane position and the actual position and is given as

$$z_n = \sqrt{(x_n^{ue} - \hat{x}_n^{ue})^2 + (y_n^{ue} - \hat{y}_n^{ue})^2}, \quad (5)$$

where x_n^{ue} and y_n^{ue} are the user x-coordinate and y-coordinate, respectively. Moreover \hat{x}_n^{ue} and \hat{y}_n^{ue} are the estimated x-coordinate and y-coordinate of the user, respectively.

For the estimation of the user position, fingerprinting with the traditional K-nearest neighbour algorithm (KNN) is considered as it is one of the most utilised algorithms for RSS-based fingerprinting [24]. In KNN, the position of the UE is estimated based on the mean of K nearest neighbours locations. The Euclidean distance is used to find the nearest neighbours from the accumulated database. After a brief optimisation of localization performance, in this work, we have defined $K = 2$. For more details on the K value estimation, please refer to [22].

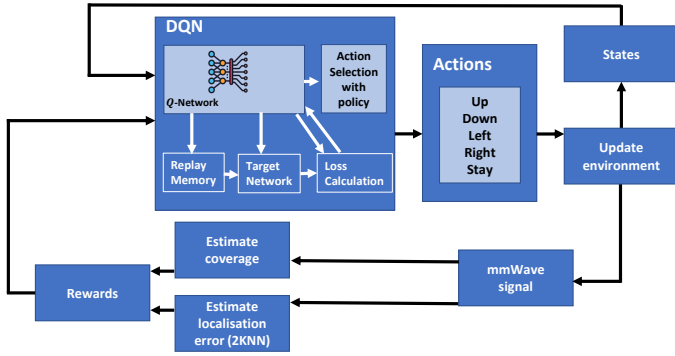


Fig. 2. BS placement optimisation with DQN.

B. Brute force algorithm (BF)

A brute force (BF) algorithm performs an exhaustive search through all the possible solutions and selects the solution that provides the best answer for the given optimisation problem. In this work, three different BF approaches are considered as a reference:

- 1) BF coverage (BFC), to maximise the coverage rate (i.e., $\max f_1(x, y)$) for the given area of interest.
- 2) BF localisation (BFL), selects the solution that minimises the localisation error (i.e., $\min f_2(x, y)$) for the area of interest.
- 3) BF joint (BFJ), selects the solution that maximises the coverage rate and minimises localisation error for the area of interest. There are multiple ways to address the general optimisation problem in (1). In this work, we chose to maximise the ratio $f_1(x, y)/f_2(x, y)$ as both $f_1(x, y)$ and $f_2(x, y)$ have different numerical ranges [25].

These BF approaches are considered in this work in order to provide a point of reference to which the performance of the proposed DQN algorithm can be compared. This is possible due to the discretisation of the space of candidate BS locations.

IV. PLACEMENT OPTIMISATION WITH DRL

In this section, we introduce the DRL model, which incorporates the proposed state representations and reward signal shaping, aimed at optimising the BS placement for the given multi-objective problem. This problem involves jointly optimising the coverage rate and localisation accuracy, under the condition that one BS has already been deployed. This represents a simplified and tractable version of a scenario where some BSs may have already been deployed, perhaps only bearing in mind the coverage rate, and new BSs are added to also include the localisation accuracy as a relevant aspect in the infrastructure design.

A. DRL Algorithm

In RL, the agent continuously learns according to the rewards or punishments obtained from interacting with the environment. The agent aims to optimise the location of BSs based on coverage area and localisation accuracy. At each time step t , the agent observes the state s_t , executes an action a_t following a policy π , then receives instant reward r_t , and

transits to the next state s_{t+1} , which together form a sequence (s_t, a_t, r_t, s_{t+1}) of Markov Decision Process (MDP) [7].

The goal of an MDP is to find the optimum policy that maximises the long term discounted rewards, given by

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots = \sum_{l=1}^{\infty} \gamma^l r_{t+l}, \quad (6)$$

where $\gamma \in [0, 1)$ denotes the discount factor for weighting the future rewards. If γ is close to 0, the RL agent will focus on actions that maximise the short-term rewards, whereas if γ is close to 1, the agent favours actions with long-term rewards. The policy π is defined as the mapping from states to actions. In Q-learning [7], the agent optimises the policy to maximise the action-value function Q , which is the expectation of the rewards G_t at the current state s_t and action a_t under policy π and can be described as

$$Q(s, a|\pi) = E_{\pi}[G_t | s_t = s, a_t = a]. \quad (7)$$

However, traditional RL can only work with problems that have a limited number of states and actions, which is not applicable in our case. Therefore, deep neural network is used as an approximator to the Q function. In addition, we apply experience replay which samples the data offline, prevents *catastrophic forgetting* and utilises the target network. At the start of training, the target network is identical to the Q-network. As training progresses, the target network's parameters are updated less frequently than those of the Q-network. This approach is adopted to provide stability to the learning process [26].

The DQN is trained to minimise the loss function given as

$$L(\theta) = \mathbb{E}[(y_t - Q(s_t, a_t; \theta))^2], \quad (8)$$

where θ is a vector and represents the DQN weights that determines the policy π . The target function y_t is given as

$$y_t = r_t + \gamma \max_a Q(s_{t+1}, a_t; \theta_{\text{target}}), \quad (9)$$

where θ_{target} is the target network weights and is copied from θ every fixed number steps.

B. DRL Agent and actions space

In the proposed DRL algorithm, the agent is the BS (we will refer to the agent BS as ABS) who is trying to find the best deployment position to improve both localisation accuracy and coverage rate given that a BS has already been deployed. The agent's actions controls the location of the ABS. Therefore, we have 5 actions taken from discrete space $A = \{up, down, left, right, stay\}$, which represents moving the current ABS location to one of the directions (in the 2D domain) or to keep the ABS in the same position.

C. Proposed state representation

A typical approach for the DRL state is to have the coordinates of the pre-deployed BS and ABS as the input state. However, such approach is not suitable as it cannot adapt to changes in the starting location of the pre-deployed BS and a new training should be performed to incorporate the change in the wireless environment. Therefore, we aim to have a single algorithm that is capable to adapt for the changes in

Algorithm 1: Obtaining the states from city layout

Input: Pre-deployed BS and ABS locations

Input: The Madrid grid city layout.

Output: Tensor $3 \times x \times y$.

- 1 Construct the 2D grid ($x \times y$)
 - 2 Construct the Building layer: give 1 for buildings, 0 otherwise.
 - 3 Construct the Pre-deployed BS layer: give 1 for already deployed BSs, 0 otherwise.
 - 4 Construct the Agent BS layer: give 1 for agent BS, 0 otherwise.
-

the location of the pre-deployed BS and capturing the signal propagation characteristics due to building shadowing. In the context of our work, we propose the state to be represented as a three-layered grid. The first layer represents the location of buildings. The second layer represents the location of pre-deployed BS and the third layer represents the location of the ABS. Thus, for the considered Madrid grid segment, the states are represented by a $3 \times 19 \times 24$ tensor (i.e., composed of 3 layers of a 19×24 grid). The grid resolution could be varied depending on the separation between the BS candidate positions, in our case 19×24 is a good trade-off between complexity and performance. Each grid layer represents the position of each individual object and defined as

- Building layer: 1 for buildings, 0 otherwise.
- Static BS layer: 1 for already deployed BSs, 0 otherwise.
- Agent BS layer: 1 for agent BS, 0 otherwise.

Algorithm 1 summarises the states extraction process.

D. Reward signal shaping

In our scenario, where the objective is to maximise the coverage rate and minimise the localisation error, it is important that the reward r_t at each time step t reflects the joint optimisation problem. To achieve this, we have chosen to maximise the ratio $f_1(x, y)/f_2(x, y)$. Therefore, our reward function r_t is a scalar and is given as

$$r_t = \frac{f_1(x, y)}{f_2(x, y)} + p, \quad (10)$$

where $f_1(x, y)$ represents the coverage area percentage obtained from (2) and $f_2(x, y)$ represents the localisation error obtained from (4). The term p corresponds to a penalty to discourage illegal actions taken by the agent, such as colliding with a building or moving outside the simulation environment. To reduce the reward when an illegal action is chosen we set $p=-0.1$, and define $p=0$ otherwise.

E. Proposed DQN training and application

The training for the proposed DQN algorithm for the ABS placement is given by Algorithm 2. The algorithm starts by initialising the parameters (Lines 1~3), followed by M episodes. Each episode includes T steps and starts with resetting the states (Line 5). For the actions selection (Lines 7 and 8), we utilise the ϵ -greedy approach to balance exploration and exploitation from previous experience. After an action a_t is performed a reward r_t is received and transit to a new

Algorithm 2: DRL-based solution for BS placement (training phase)

Input: Tensor representation of the environment obtained from Algorithm 1

Output: Final ABS location.

- 1 Initialise the replay memory D to a maximum capacity
 - 2 Initialise Q-network with random weights θ
 - 3 Initialise target network with weights $\theta_{target} = \theta$
 - 4 **for** $episode = 1, \dots, M$ **do**
 - 5 Initialise the environment and receive initial state s_t
 - 6 **for** $step t = 1, \dots, T$ **do**
 - 7 with probability ϵ select random action a_t
 - 8 otherwise select $a_t = \max_a Q(s_t, a_t; \theta)$
 - 9 observe r_t and new state s_{t+1}
 - 10 store transition $\{s_t, a_t, r_t, s_{t+1}\}$ in D
 - 11 **if** *memory is full* **then**
 - 12 sample mini-batch randomly of transitions $\{s_t, a_t, r_t, s_{t+1}\}$ in D
 - 13 set $y_t = \begin{cases} r_t & t = T \\ r_t + \gamma \max_a Q(s_t, a_t; \theta) & t < T \end{cases}$
 - 14 update weights for θ of the main Q-network by minimising the loss function (8)
 - 15 set $\theta_{target} = \theta$ every τ steps
-

state s_{t+1} . The transition tuple $\{s_t, a_t, r_t, s_{t+1}\}$ is stored in experience replay memory D , which stores experiences in a first-in-first-out manner (Line 10). Once D is larger than the mini-batch size, the network training starts (Lines 11~15). Random transition tuples of mini-batch size are sampled from D to train the DQN from past experiences. The target network is used to estimate the target value y_t (Line 13), which is used to evaluate the actions selected by the main Q-network. The loss function is found from (8), which is used to update the main Q-network parameters θ (Line 14). The target network parameters θ_{target} are updated every fixed number τ of training steps (Line 15).

Once the training is finished, the proposed DQN learns the placement of the ABS. During application, the trained DQN observes the environment state s_t at each step and selects an action that maximises $f_1(x, y)/f_2(x, y)$. This is repeated multiple times (50 in our case) until the optimum ABS location is found.

V. PERFORMANCE ANALYSIS

A. Evaluation scenario

In the DRL training stage, the DQN model is trained for 3000 episodes with 200 steps per episode. Adam optimiser with adaptive learning rate (L_r) was used for the training with $L_r=10^{-3}$ for the first 500 episodes, $L_r=10^{-4}$ for the following 500 episodes and $L_r=10^{-5}$ for the rest of episodes. Reducing L_r allows the optimiser to find the minimum in the loss more efficiently [27]. Parameter γ is set to 0.9, the mini-batch size is set to 64, the memory buffer is set to 20000, the target network update frequency τ is set to 50 and

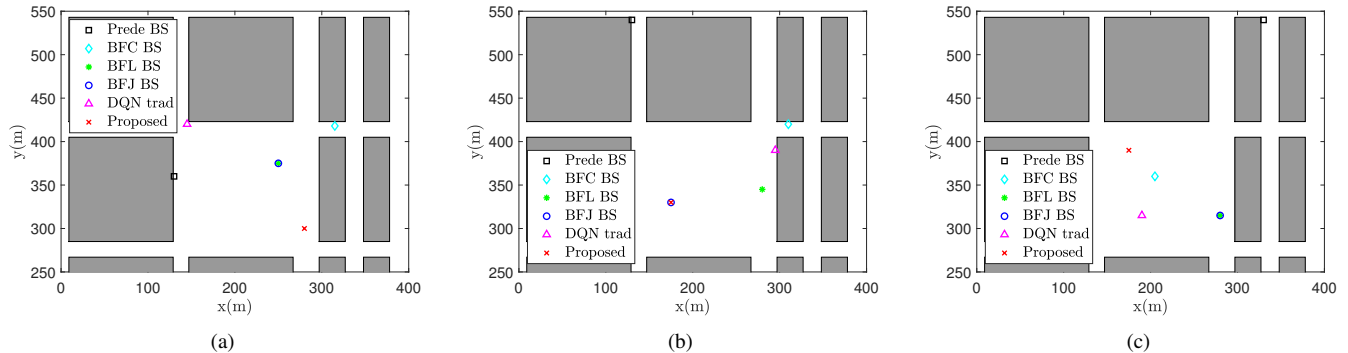


Fig. 3. 2D visualization for the BS locations. (a) Case 1, (b) Case 2, (c) Case 3.

TABLE I
COVERAGE RATE AND LOCALISATION ERROR (IN METER).

Pre-deployed BS	BFC			BFL			BFJ			Traditional DQN			Proposed DQN		
	Cov rate	Loc error	Cov rate / Loc error	Cov rate	Loc error	Cov rate / Loc error	Cov rate	Loc error	Cov rate / Loc error	Cov rate	Loc error	Cov rate / Loc error	Cov rate	Loc error	Cov rate / Loc error
1	0.98	2.45	0.40	0.90	1.46	0.61	0.90	1.46	0.61	0.88	2.19	0.40	0.92	1.73	0.53
2	0.95	3.37	0.28	0.86	1.97	0.43	0.87	1.97	0.44	0.90	3.21	0.28	0.87	1.97	0.44
3	0.98	2.17	0.45	0.95	1.71	0.55	0.95	1.71	0.55	0.95	2.35	0.40	0.98	1.87	0.52

the Mean Squared Error loss function is used. In our study, we compare two DQNs: our proposed version with a grid-based state representation and the traditional version that uses a coordinate-based state representation for both pre-deployed and new Base Stations (BS). The traditional DQN model is structured with a Q-network that consists of two hidden layers containing 50 and 25 neurons, respectively. These layers utilize the ReLU activation function for the hidden layers and a Linear activation function for the output layer. In contrast, our proposed DQN model integrates two CNN layers on top of these hidden layers. The kernel size of these CNN layers is 4×5 , and there is a 2×2 max pooling layer following the first CNN layer. Additionally, the starting location of our ABS is randomly determined at the start of each training episode. To evaluate the model's effectiveness, 70% of the available pre-deployed BSs are used for training, and the remaining 30% are utilized for testing. We set the threshold for the received signal strength (δ) to -80 dBm.

B. Numerical results

The achieved localisation error and coverage rate are shown in Table I for the considered algorithms (i.e., BFC, BFL, BFJ, traditional DQN and proposed DQN) for three different pre-deployed BS scenarios. Fig. 3 shows a 2D visualization of the BS placement for the different approaches. The BSs deployment scenarios order in Table I matches what is shown in Fig. 3. As it can be appreciated from Table I, BFC finds the best location to maximise the coverage rate, but it has a significant impact on the localisation accuracy. Taking Case 2 as an example, the BFC coverage rate is 95% while the localisation error is more than 3.37 m. On the other hand, the smallest localisation error given by BFL is 1.97 m at the expense of reducing the coverage rate to 86%.

The aim of the traditional and proposed DQN algorithm is to provide a solution that is similar to the one obtained from BFJ.

Therefore, we also investigate the Cov rate/Loc error ratio. The three shown cases are taken from the test data (locations of pre-deployed BS that are not used while training the DQN). As it can be seen from Table I, the proposed DQN performance either matches (Case 2) or provides a similar (Cases 1 and 3) coverage and localisation error as BFJ. Therefore, the capability of the proposed DQN to adapt to the changes in the location of the deployed BS is demonstrated, while the traditional DQN approach fails to adapt. The reason why the results for the proposed DQN are not identical to BFJ is as we only trained for 70% of the possible pre-deployed BSs and the shown results come from the testing (i.e., not seen by the model during training). Moreover, the proposed DQN model needs to be trained only once while BFJ needs to be trained for each scenario. Note that the coverage rate and localisation error achievable by the proposed DQN are not higher than those of BFC and BFL, respectively, as they represent the optimal solutions for coverage rate and localisation error when considered independently. In contrast, the DQN aims for a solution that is jointly optimized.

Finally, Fig. 4 shows the effect of selecting different ABS locations and its effect on both localisation error and coverage rate. The pre-deployed BS location is the same as shown in Fig. 3(a). The localisation error and coverage rate have different optimal locations and the ABS location to optimise the coverage rate is not the same for optimising the localisation error. In other words, there is a trade-off between the two optimisation problems in BS location selection that needs to be addressed through network planning, depending on which parameter is more significant in each specific scenario.

VI. CONCLUSIONS

This work has investigated the placement optimisation of mmWave BS in the presence of a pre-deployed BS to simultaneously optimise the coverage rate and localisation accuracy

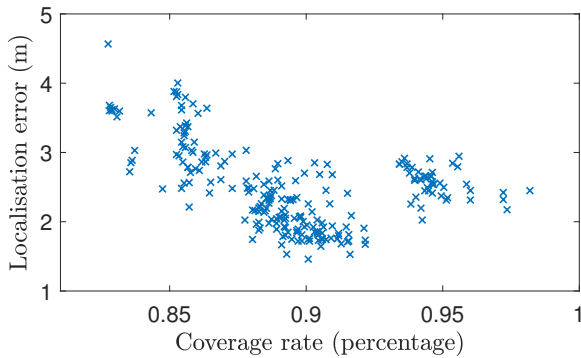


Fig. 4. Localisation error vs coverage rate.

in an urban city layout. We presented a DRL algorithm that is capable to solve the multi-objective problem and adapt to the changes in the location of the pre-deployed BS by proposing a three-layered state representation that is capable to capture spatial properties of the radio environment. Numerical results have demonstrated that the proposed algorithm provides similar results as the optimum exhaustive search algorithms. The reason why the results for the proposed DQN are not identical to BFJ is as we only trained for 70% of the possible pre-deployed BSs and the shown results come from the testing. Nevertheless, the proposed DQN model needs to be trained only once while BFJ needs to be trained for each scenario. In addition, this work has demonstrated that there is a trade-off between localisation accuracy and coverage rate. Future work will extend the current work to a multi-agent scenario.

ACKNOWLEDGMENT

This work was supported by the Academy of Finland Flagship program: Finnish Center for Artificial Intelligence FCAI and the Academy of Finland project 347197 Artificial Intelligence for Urban Low-Emission Autonomous Traffic (AIforLEssAuto).

REFERENCES

- [1] A. Ajanovic, R. Haas, and F. Wirl, "Reducing CO2 emissions of cars in the EU: analyzing the underlying mechanisms of standards, registration taxes and fuel taxes," *Energy Efficiency*, vol. 9, 08 2016.
- [2] P. Belanovic, D. Valerio, A. Paier, T. Zemen, F. Ricciato, and C. F. Mecklenbrauker, "On wireless links for vehicle-to-infrastructure communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 269–282, 2010.
- [3] 3GPP, *Service requirements for the 5G system*. 3rd Generation Partnership Project (3GPP), Technical Specifications (TS) 22.261, 12, version 18.1.0., 2016.
- [4] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4391–4403, 2013.
- [5] W. K. Alsaedi, H. Ahmadi, Z. Khan, and D. Grace, "Spectrum options and allocations for 6G: A regulatory and standardization review," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 1787–1812, 2023.
- [6] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of cellular mobile radio localization methods: From 1G to 5G," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1124–1148, 2018.
- [7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018. [Online]. Available: <http://incompleteideas.net/book/the-book-2nd.html>
- [8] H. Kim, T. Fujii, and K. Umebayashi, "Relay nodes selection using reinforcement learning," in *2021 International Conference on Artificial*

- Intelligence in Information and Communication (ICAIC)*, 2021, pp. 329–334.
- [9] S. S. Oyewobi, G. P. Hancke, A. M. Abu-Mahfouz, and A. J. Onumanyi, "An effective spectrum handoff based on reinforcement learning for target channel selection in the industrial internet of things," *Sensors*, vol. 19, no. 6, 2019.
- [10] M. S. Frikha, S. M. Gammar, A. Lahmadi, and L. Andrey, "Reinforcement and deep reinforcement learning for wireless internet of things: A survey," *Computer Communications*, vol. 178, pp. 98–113, 2021.
- [11] S. S. Szyszkowicz, A. Lou, and H. Yanikomeroglu, "Automated placement of individual millimeter-wave wall-mounted base stations for line-of-sight coverage of outdoor urban areas," *IEEE Wireless Communications Letters*, vol. 5, no. 3, pp. 316–319, 2016.
- [12] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Efficient millimeter-wave infrastructure placement for city-scale ITS," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, 2019, pp. 1–5.
- [13] J. Zhang, Q. Wang, and H. Ahmadi, "An integrated access and backhaul approach to sustainable dense small cell network planning," *Information*, vol. 15, no. 1, 2024. [Online]. Available: <https://www.mdpi.com/2078-2489/15/1/19>
- [14] J. Wu, P. Yu, L. Feng, F. Zhou, W. Li, and X. Qiu, "3D aerial base station position planning based on deep Q-network for capacity enhancement," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 482–487.
- [15] G. B. Tarekegn, R.-T. Juang, H.-P. Lin, Y. Y. Munaye, L.-C. Wang, and M. A. Bitew, "Deep-reinforcement-learning-based drone base station deployment for wireless communication services," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 899–21 915, 2022.
- [16] S. P. Gopi and M. Magarini, "Reinforcement learning aided UAV base station location optimization for rate maximization," *Electronics*, vol. 10, no. 23, 2021.
- [17] J. Qiu, J. Lyu, and L. Fu, "Placement optimization of aerial base stations with deep reinforcement learning," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.
- [18] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) TS 38.901, 01 2020, version 16.1.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173>
- [19] V. Nurmela *et al.*, "Deliverable D1.4: METIS channel models." Proc. Mobile Wireless Commun. Enablers Twenty-Two Inf. Soc. (METIS), 2015, p. 1.
- [20] Remcom. Wireless InSite - 3D wireless prediction software. Accessed: Jan 27, 2021. [Online]. Available: <https://www.remcom.com/wireless-insite-em-propagation-software>
- [21] R. Klus, L. Klus, D. Solomitskii, J. Talvitie, and M. Valkama, "Deep learning-based cell-level and beam-level mobility management system," *Sensors*, vol. 20, no. 24, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/24/7124>
- [22] A. Al-Tahmeesschi, J. Talvitie, M. López-Benítez, and L. Ruotsalainen, "Deep learning-based fingerprinting for outdoor UE positioning utilising spatially correlated RSSs of 5G networks," in *2022 International Conference on Localization and GNSS (ICL-GNSS)*, 2022, pp. 1–7.
- [23] 3GPP, "NR; Physical layer measurements," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.215, 07 2020, version 15.7.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3217>
- [24] M. Y. Umair, K. V. Ramana, and Y. Dongkai, "An enhanced K-Nearest Neighbor algorithm for indoor positioning systems in a WLAN," in *2014 IEEE Computers, Communications and IT Applications Conference*, 2014, pp. 19–23.
- [25] E. Björnson, E. A. Jorswieck, M. Debbah, and B. Ottersten, "Multiobjective signal processing optimization: The way to balance conflicting metrics in 5G systems," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 14–23, 2014.
- [26] B. Brown and A. Zai, *Deep Reinforcement Learning in Action*. Manning Publications, 2020.
- [27] K. He, R. Girshick, and P. Dollar, "Rethinking ImageNet pre-training," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4917–4926.