Sequence Q-Learning Algorithm for Optimal Mobility-Aware User Association

Wanjun Ning*, Zimu Xu*, Jingjin Wu*, Tiejun Tong†

*Department of Statistics, BNU-HKBU United International College, Zhuhai, Guangdong, P. R. China
†Department of Mathematics, Hong Kong Baptist University, Kowloon, Hong Kong SAR, P. R. China
Email: q030202601@mail.uic.edu.cn; r130202603@mail.uic.edu.cn; jj.wu@ieee.org; tongt@hkbu.edu.hk

Abstract—We consider a wireless network scenario applicable to metropolitan areas with developed public transport networks and high commute demands, where the mobile user equipments (UEs) move along fixed and predetermined trajectories and request to associate with millimeter-wave (mmWave) base stations (BSs). An effective and efficient algorithm, called the Sequence Q-learning Algorithm (SQA), is proposed to maximize the long-run average transmission rate of the network, which is an NP-hard problem. Furthermore, the SQA tackles the complexity issue by only allowing possible re-associations (handover of a UE from one BS to another) at a discrete set of decision epochs and has polynomial time complexity. This feature of the SQA also restricts too frequent handovers, which are considered highly undesirable in mmWave networks. Moreover, we demonstrate by extensive numerical results that the SQA can significantly outperform the benchmark algorithms proposed in existing research by taking all UEs' future trajectories and possible decisions into account at every decision epoch.

Index Terms—Mobility-aware user association, NP-hard optimization, Sequence Q-learning Algorithm, Reinforcement learning, mmWave communication

I. INTRODUCTION

In next-generation wireless networks, access points such as mobile base stations (BSs) are densely deployed to provide redundant coverage and fault tolerance if some of the BSs fail. As a result, mobile user equipments (UEs) can choose from multiple BSs for accessing the core network. Studies on mobility-aware user association, which focus on choosing the proper BS for each UE to improve network performance as reflected by certain Quality of Service (QoS) metrics, have thus become a popular research topic [1], [2].

Operating on frequency bands between 30 and 300 GHz, mmWave communications can provide much higher transmission rates than communications on sub-6GHz bands [3], [4]. However, the susceptibility to physical obstacles for transmissions on mmWave bands creates new issues for research on user association strategies [3], [5]. Meanwhile, a typical mmWave BS covers a much smaller area than a conventional sub-6GHz BS, leading to more frequent re-associations (handovers between different BSs) [6]. As handovers incur extra power consumption and are more likely to result in connection failures, studies on user association strategies in a mobility-aware mmWave environment must account

for the number of handovers in addition to the traditional objectives and constraints.

Most existing studies on mobility-aware user association consider that UEs move along random trajectories following certain statistical distributions [5], [7], [8]. However, as most actual UE movements cannot fit in theoretical distributions [9], the practical value of strategies under this assumption is limited.

This paper focuses on a scenario applicable for cities with developed public transport networks and high commute demands. Statistics show that more than 45% of people choose public transport for commuting to and from work in metropolises such as Singapore, London, and Hong Kong [10]. For mobile users taking subways or public buses, it is possible to estimate their UEs' future movements and positions rather accurately for a relatively long time. Due to the large proportion of such users and their usage demands, it is reasonable to propose a user association strategy for the UEs that move along a predetermined trajectory.

The future trajectories of such UEs could be exploited by user association strategies to improve the performance further. More specifically, we will only allow a UE to handover from its current serving BS to another BS at specific points on its trajectory when certain triggering conditions are met. This approach could avoid frequent handovers and simultaneously reduce the complexity of the user association algorithm, as association decisions only need to be made at a discrete set of *decision epochs* when the triggering conditions are potentially fulfilled. On the other hand, given that a BS's available bandwidth is limited, it is beneficial to take other UEs' possible future actions based on their trajectories into account to improve the long-run performance.

Our proposed approach to exploit the future information is applying the reinforcement learning (RL) technique. RL is a machine learning technique that learns system information from the interaction between agents and the environment to solve decision-making problems such as user association problems. For example, a deep RL (DRL)-based algorithm, Deep Q-Network, was demonstrated in [11] to make user association decisions regardless of mobility of UEs. A distributed method called multi-agent DQN with recurrent neural networks

was proposed in [12] to optimize user association decisions, considering channel dynamics and changing rate demand of UEs. Guo *et al.* [13] propose a multi-agent proximal policy optimization to solve the joint optimization problem of handover control and power allocation in two-layer heterogeneous cellular networks and show it is effective in small-scale experiments. Sun *et al.* [8] demonstrated a handover strategy based on RL called SMART, considering mmWave channel characteristics and UE's QoS requirements. A Q-Learning-based handover strategy was proposed in [14], which maximized the trajectory rate but ignored the interactions between UE decisions.

The new algorithm proposed in this work is called the Sequence Q-learning Algorithm (SQA). It aims at maximizing the long-run average transmission rate of the network, a commonly used QoS metric. In particular, the SQA considers interactions between different UEs based on the predicted trajectories, which traditional Q-learning could not account due to the curse of dimensionality of the state space. Besides, the SQA can efficiently make use of future information and achieve significant performance improvement in moderate scale experiments, in which DRL approaches (e.g. [13]) require extremely massive training steps and hardly converge.

The main contribution of this paper is to propose an efficient SQA for mobility-aware user association, aiming at optimizing the long-run average transmission rate of the network while limiting the number of re-association to a relatively low level. The SQA is specifically designed for the scenario where the trajectories of UEs are known or can be reasonably predicted by the mobile service operators. By taking advantage of the future movement of UEs, the SQA explores appropriate weights for each possible association action based on the classical Q-learning. The weights reflect the possible impact of future movements and decisions of all UEs and are included in the adjusted action-value functions to identify the optimal BS associated with a moving UE at each decision epoch. We will demonstrate that, compared with state-of-the-art benchmark algorithms, the SQA achieves much better performance in terms of the long-run average transmission rate of the network.

The remainder of this paper is organized as follows. Section II describes the system model and formulates the optimization problem. Section III presents the proposed SQA in detail. We describe the experimental setup and present the numerical results in Section IV, where our proposed algorithm and four state-of-the-art benchmark algorithms are compared. Section V concludes the paper.

II. System Model and Problem Formulation

Let $\mathscr{M} = \{1, \dots, M\}$ denote the set of BSs, and $\mathscr{N} = \{1, \dots, N\}$ represent the set of UEs. Accordingly, there are in total M BSs and N UEs randomly and uniformly distributed in the system. We denote $dis_{m,n}(t)$,

where $m \in \mathscr{M}$ and $n \in \mathscr{N}$, as the Euclidean distance between BS m and UE n at time t. A UE is associated to one BS at any time.

Consider $\mathscr{D}(n, t)$ as the set of candidate BSs available to be associated by UE n at time t. As frequent handover of UEs is considered undesirable for mobile networks [6], the re-association would only occur at decision epochs when certain triggering conditions are met. Considering the noise-limited nature of mmWave network [15], we associate the triggering conditions with the Signal-to-Noise Ratio (SNR). Specifically, for a UE, decision epochs are the moments when the SNR from its currently serving BS falls lower than a certain threshold, or when the SNR from a neighboring BS becomes higher than a certain threshold [16]. Assuming that such changes do not occur concurrently for more than one UE, $\mathscr{T} = \{t_1, t_2, \dots\}$ is denoted as the set of decision epochs, where each $t_i \in \mathscr{R}$ is a specific time point when a change in $\mathscr{D}(n, t_i)$ occurs for a UE $n \in \mathscr{N}$. We assume $t_i < t_j$ for all $i < j$, thus a decision epoch t_i uniquely identifies the UE n that experiences a change in $\mathscr{D}(n, t_i)$. We denote this relationship as $n = U(t_i)$. For notational simplicity, we define $D(t_i) := \mathscr{D}(n, t_i)$ as the decision set at decision epoch t_i.

As the previous section explains, we consider a scenario where all UEs move along predetermined trajectories. Accordingly, when user association decisions are made at t_i, the future locations of UEs at following times can be taken into account. We further assume that the latency required for a UE to obtain the states of all BSs in the network is negligible compared to the time gap between any two consecutive decision epochs, such that the network state may only change at decision epochs.

Two important evaluation metrics in optimization problems in user association are the long-run average transmission rate and the number of handovers. While the number of handovers is controlled by only allowing potential handovers at decision epochs, we now focus on maximizing the long-run average rate of transmission.

To formulate the optimization problem, we define another function, $last(t)$, that returns the latest decision epoch $t_i \in \mathscr{T}$ before time t. Notably, $last(t_i)$, where t_i itself is a decision epoch, will return the last decision epoch t_{i-1}. Recall that $U(t)$ returns the corresponding UE $n \in \mathscr{N}$ whose set of candidate BSs is changed at t. In this way, for any t, we can identify a unique BS, $A_i \in D(t_i)$, where $t_i = last(t)$, that UE $U(t)$ decides to associate with at t_i. We further denote $cnt_m(t)$ as the number of UEs associated with BS m at t.

We further assume that the bandwidth of a BS is evenly distributed to all associating UEs. The transmission rate of UE n associated with BS m at time t is

$$r_{m,n}(t) = \frac{B_m}{cnt_m(t)} \log_2 \left(1 + \frac{P_{m,n} dis_{m,n}(t)^{-\beta}}{N_0} \right), \quad (1)$$

where $P_{m,n}$ is the transmission power for BS m to com-

municate with UE n, β is the path-loss exponent, B_m is the bandwidth of BS m, and N_0 is the thermal noise. The long-run average transmission rate is defined as

$$\bar{L} = \lim_{t \to \infty} \frac{1}{t} \int_0^t \sum_{n \in \mathcal{N}} O_{m,n}(t) r_{m,n}(t), \tag{2}$$

where $O_{m,n}(t) = 1$ if UE n is associated to BS m at t and $O_{m,n}(t) = 0$ otherwise. It is straightforward to observe that all $O_{m,n}(t)$ values are determined by the association decision A_i taken before t.

Our optimization problem can be formulated as

$$\begin{aligned}
\max \quad & \bar{L} \\
\text{s.t.:} \quad & \sum_{m \in \mathcal{M}} O_{m,n}(t) = 1, \quad \forall n \in \mathcal{N}, t > 0; \\
& O_{m,n}(t) - O_{m,n}(\text{last}(t)^-) = 0, \\
& \quad \forall n \neq U(t), \forall m \in \mathcal{M}, t > 0; \\
& \sum_{m \notin D(\text{last}_t)} O_{m,n}(t) = 0, \quad \text{for } n = U(t), t > 0; \\
& \text{cnt}_m(t) - \sum_{n \in \mathcal{N}} O_{m,n}(t) = 0, \quad \forall m \in \mathcal{M},
\end{aligned} \tag{3}$$

where $\text{last}(t)^-$ refers to the moment just before $\text{last}(t)$.

Note that this problem is an NP-hard problem. We provide a sketch of the proof. A sufficient condition to prove that a problem is NP-hard is that a known NP-hard problem can be reduced to the concerned problem [17]. In our case, it has been shown in existing research that the static user association optimization problem (the static UA problem), which focuses on associating UEs to BSs in a single moment to maximize network performance or minimize total cost, is NP-hard [18]. In this sense, to show that problem (3) is NP-hard, we need to show that (3) is reducible to the static UA problem, such that if a black box solver S can solve the problem (3), S should be able to solve all static UA problems.

We then consider a virtual scenario to show the connection between problem (3) and the static UA problem. Note that this virtual scenario is the only demonstration to show that problem (3) is NP-hard and does not apply to the actual implementation of our proposed algorithm. First, we define a virtual BS v, which all UEs are associated with at $t = 0$ but will not be available for the association for any UE that has already connected to another BS. We further assume that all UEs are located near the (virtual) boundary of v at $t = 0$. Then, we move every UE sequentially until they cross the boundary and trigger a virtual decision epoch. In this sense, the time gap between consecutive virtual decision epochs would be very short, and all UEs would be associated with a non-virtual BS by $t = \delta$, where δ is an arbitrarily small positive actual number. Suppose our objective is to maximize the transmission rate, the above problem could be formulated either as a static UA (as each UE would only re-associate once) to maximize the instant

transmission rate or as an instance of the problem (3) to maximize the average transmission rate from $t = 0$ to $t = \delta$. Again, as the static UA problem is known to be NP-hard, problem (3) is also NP-hard. Due to the space limit, we have proven the NP-hard only by a simplified scenario; yet a complete proof for more general settings can also be readily conducted.

III. SEQUENCE Q-LEARNING ALGORITHM

We now define necessary notations and clarify related concepts to describe the Sequence Q-learning Algorithm (SQA) proposed to solve the problem (3) in detail.

Recall that, for the decision epochs from t_1 up to t_j, we have a tuple of decisions $A_1, A_2, \cdots, A_j$, with each decision A_i representing the BS associated by UE $U(t_i)$ at decision epoch t_i. Define an operation $\oplus$ as attaching an element to the end of a tuple, namely $(x_1, x_2, ..., x_n) \oplus x_{n+1} = (x_1, x_2, ..., x_n, x_{n+1})$.

We further denote $S_k = (A_1, A_2, ..., A_{k-1})$ as the set of decisions that have been made before the decision epoch t_K. Our aim can be redefined as identifying the best decision series $(A_k, A_{k+1}, ..., A_T | S_k)$ at every t_k. Therefore, problem (3) can be considered as a Markov Decision Process (MDP). For any decision series S_k, we further use $C_i^{(k)}$ to represent the set of all possible decision series up to the decision epoch t_i, where $i > k$. That is, $C_i^{(k)} = \{S_k \oplus d_k \oplus d_{k+1} \oplus ... \oplus d_{i-1} : d_j \in D(t_j)\}$. For convenience, $C_i^{(k)}$ represents an ordered set sorted by the index of d_k, then by d_{k+1}, and so on. We then denote the jth element in $C_i^{(k)}$ as $c_{i,j}^{(k)}$. Note that the number of elements in $C_i^{(k)}$ is $\sum_{j=1}^{\prod_{l=k}^{i-1} |D_l|}$.

As problem (3) can be considered as an MDP, it is straightforward to attempt to solve it by the RL techniques. As one of the classical RL techniques, Q-learning performs well in MDPs with low dimension discrete action space [19]. However, the relatively high-dimensional state space of problem (3) prohibits the direct application of the classical Q-learning method. To address this issue, we propose the SQA based on Q-learning. While the SQA retains the advantages of Q-learning, it is much better in handling high-dimensional state space problems, as we will demonstrate later.

In the classical Q-learning [20], the optimal action-value function for an action a, when the current state is s, is defined as

$$\begin{aligned}
Q^*(s,a) = \sum_{s'} P\left(s' \mid s,a\right) \Big[& R\left(s' \mid s,a\right) \\
& + \gamma \cdot \max_{a'} Q^*\left(s', a'\right) \Big],
\end{aligned} \tag{4}$$

where $P(s' \mid s,a)$ is the transition probability that the next state is s' when taking action a under current state s, $R(s' \mid s,a)$ is the reward associated with the action and the transition, and $\gamma \in (0,1)$ is the discount rate.

Similar to (4), we use $Q_i^*(S_k, A_i)$ to indicate the optimal action-value function under given state history S_k, which can be achieved by connecting UE $U(t_i)$ to BS A_i at decision epoch t_i. Specifically,

$$Q_i^*(S_k, A_i) = E_{\mathbf{w}}\left[Q_i^*(S_i, A_i)\right] = \sum_{j=1}^{\Pi_{l=k}^{i-1}|D_l|} w_{i,j} Q_i^*\left(c_{i,j}^{(k)}, A_i\right)$$

$$= \sum_{j=1}^{\Pi_{l=k}^{i-1}|D_l|} w_{i,j}\left(R_{c_{i,j}^{(k)}, A_i} + \gamma \max_{A_{i+1}} Q_{i+1}^*\left(c_{i,j}^{(k)} \oplus A_i, A_{i+1}\right)\right)$$

$$= \sum_{j=1}^{\Pi_{l=k}^{i-1}|D_l|} w_{i,j} R_{c_{i,j}^{(k)}, A_i} + \gamma \max_{A_{i+1}} Q_{(i+1)|i}^*(S_k, A_{i+1} \mid A_i)$$

(5)

where $\mathbf{w}$ is the weighed matrix, $w_{i,j} \in \mathbf{w}$ is the weight of $c_{i,j}^{(k)}$, γ is the discount rate, and

$$Q_{(i+1)|i}^*\left(S_k, A_{i+1} \mid A_i\right)$$
$$= \sum_{j=1}^{\Pi_{l=k}^{i-1}|D_l|} w_{i,j} Q_{i+1}^*\left(c_{i,j}^{(k)} \oplus A_i, A_{i+1}\right).$$

(6)

For each decision epoch i, if $Q_i^*(S_k, d_{i,j}) > Q_i^*(S_k, d_{i,l})$, we claim that BS candidate $d_{i,j}$ is better than candidate $d_{i,l}$, where $d_{i,j}, d_{i,l} \in D_i$. Thus, if we claim a BS candidate b is good, it shall have a relatively high $Q_i^*(S_k, b)$ among all candidates in D_i. Ideally, there exists an appropriate $\mathbf{w}$ that makes $Q_i^*(S_k, d_{i,j}) > Q_i^*(S_k, d_{i,l})$ if $d_{i,j}$ is chosen in global optimal solution and $j \neq l$. We can construct the global optimal solution with the previous instructions by connecting the corresponding UE to the best BS candidate at each decision epoch.

Unfortunately, because it is not realistic to enumerate all the possible decision series up to decision epoch i, we cannot obtain an ideal and true Q table and $\mathbf{w}$. Thus, we use $\hat{Q}$ to approximate the Q^* table and $\hat{\mathbf{w}}$ to approximate $\mathbf{w}$. It is straightforward to vertify that

$$\hat{Q}_i(S_k, A_i) = E_{\hat{\mathbf{w}}}\left[\hat{Q}_i(S_i, A_i)\right]$$
$$= \sum_{j=1}^{\Pi_{l=k}^{i-1}|D_l|} \hat{w}_{i,j} R_{c_{i,j}^{(k)}, A_i} + \gamma \max_{A_{i+1}} \hat{Q}_{(i+1)|i}(S_k, A_{i+1} \mid A_i).$$

(7)

Since it is hard to directly give a reasonable $\hat{\mathbf{w}}$, while implementing the algorithm to calculate $\hat{Q}$, we let $\hat{\mathbf{w}}$ be obtained from exploration strategy instead of assigning fixed values to $\hat{\mathbf{w}}$.

To show how to obtain $\hat{\mathbf{w}}$ by the exploration strategy, we need to formulate the exploration process. The transition probability $p_{i,j}$ is introduced for describing the exploration strategy, i.e., the probability of connecting corresponding UE to BS candidate $d_{i,j}$ at the decision

epoch $t_i \in T$ ($p_{i,j} = 0$ if $d_{i,j}$ is not feasible). Then, we can formulate the exploration process as

$$\hat{Q}_{i+1}(S_k, A_{i+1}) = \sum_{j=1}^{|D_i|} p_{i,j} \hat{Q}_{(i+1)|i}(S_k, A_{i+1} \mid d_{i,j})$$
$$= \sum_{j=1}^{\Pi_{l=k}^{i}|D_l|} \hat{w}_{i,j} \hat{Q}_{i+1}\left(c_{i+1,j}^{(k)}, A_{i+1}\right).$$

(8)

Referring to the concept of ideal $\mathbf{w}$ we mentioned before, we can infer that a good approximation of the ideal $\mathbf{w}$ should give high weight to those real good BS candidates at each decision epoch, which means $\hat{Q}_{(i+1)|i}(S_k, A_{i+1})$ should be dominated by $\hat{Q}_{i+1}\left(c_{i,j}^{(k)} \oplus A_i, A_{i+1}\right)$ of good A_i. Thus, in algorithm implementation, we can use the Monte Carlo method to sample good decision series and use their return (the concept in MDP) to estimate $\hat{Q}_{(i+1)|i}(S_k, A_{i+1})$.

To get good decision series and $\hat{\mathbf{w}}$, we need an exploration strategy that assigns a high transition probability to good BS candidates at each decision epoch. Specifically, the transition probability $p_{i,j}$ is defined as

$$p_{i,j} = \frac{\epsilon^{\phi_i(j)}}{\sum_{l=1}^{|D_i|} \epsilon^{\phi_i(l)}},$$

(9)

where ϵ is a preset hyper parameter, and $\phi_i(j)$ is the number of actions in D_i whose $\hat{Q}$ value is less than $\hat{Q}_i(S_k, d_{i,j})$. Then, we can design the update rule as

$$\hat{Q}_i(S_k, A_i) \leftarrow \hat{Q}_i(S_k, A_i) + \alpha\gamma \max_{A_{i+1}} \hat{Q}_{(i+1)|i}(S_k, A_{i+1})$$
$$- \alpha\hat{Q}_i(S_k, A_i) + R_{S_i},$$

(10)

where α is the learning rate. R_{S_i} is the sum transmission rate between t_i to t_{i+1}. Assume that decision series $s_i \in c_i^{(k)}$ are made the same, then

$$R_{s_i} = \sum_{l=1}^{N} \int_{t_{i+1}-t_i}^{t_{i+1}-t_i} r(t)\Delta t.$$

(11)

The pseudo-code of the SQA is presented in Algorithm 1. In Algorithm 1, we set a parameter *STEP* to indicate the maximum number of steps for exploration from any decision epoch, which could control the running time of the SQA. By adjusting the values of *STEP* and the discount rate γ, we may fine-tune the performance and running time according to specific network scenarios and UE trajectories.

Next, we illustrate how we apply the SQA to solve (3), with the flowchart shown in Fig. 1. It is worth noting that SQA is called between two decision epochs instead of after each new request. In this way, the association decision is made immediately after each request without any delay because the decision can be made by only checking the $\hat{Q}$ table.

Algorithm 1 Sequence Q-Learning

Input: $\hat{Q}_i\left(S_{k-1}, d_{i,j}\right),\ D_i,\ S_{k-1}$
$\qquad\qquad i \in \{k, k+1, ..., T\}, d_{i,j} \in D_i$
Output: $\hat{Q}_i\left(S_k, d_{i,j}\right) \quad i \in \{k, k+1, ..., T\}$

1: **for** $i = k$ to T **do**
2: $\quad \hat{Q}_i\left(S_k, d_{i,j}\right) \longleftarrow \hat{Q}_i\left(S_{k-1}, d_{i,j}\right)$
3: **end for**
4: **for** $j = 0$ to max-iteration **do**
5: $\quad step \longleftarrow 0$
6: $\quad$ Explore (S_i)
7: $\quad$ update $p_{i,j}$ by $\hat{Q}_i\left(S_k, d_{i,j}\right)$
8: **end for**
9:
10: **Procedure** Explore $(S_i :$ decision series up
$\quad$ to decision epoch $i)$
11: **if** $(i == T)$ or $(step > STEP)$ **then**
12: $\quad$ **return** 0
13: **end if**
14: choose BS A_i by p_i
15: $step \longleftarrow step + 1$
16: $S_{i+1} \longleftarrow S_i \oplus A_i$
17: $\text{return}_{\text{MDP}} \longleftarrow \text{R}_{S_{i+1}} + \gamma \text{Explore}(S_{i+1})$
18: $\hat{Q}_{(i+1)|i}\left(S_k, A_{i+1}\right) \longleftarrow \hat{Q}_{(i+1)|i}\left(S_k, A_{i+1}\right)$
$\qquad\qquad + \alpha\left[\text{return}_{\text{MDP}} - \hat{Q}_{(i+1)|i}\left(S_k, A_{i+1}\right)\right]$
19: $\hat{Q}_i\left(S_k, A_i\right) \longleftarrow \hat{Q}_i\left(S_k, A_i\right) + \alpha\left[\text{R}_{S_{i+1}} - \hat{Q}_i\left(S_k, A_i\right)\right.$
$\qquad\qquad \left. + \gamma \max_{A_{i+1}} \hat{Q}_{(i+1)|i}\left(S_k, A_{i+1}|A_i\right)\right]$
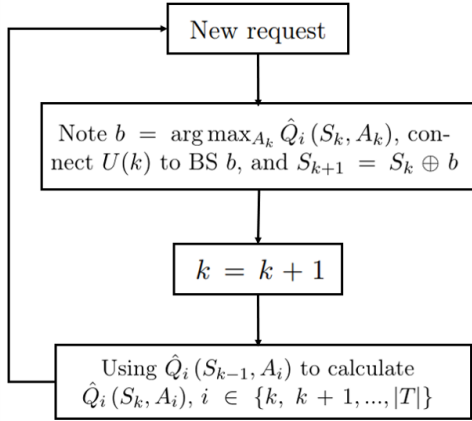20: **return** $\text{R}_{S_{i+1}} + \gamma \text{return}_{\text{MDP}}$

Fig. 1. Flowchart of applying Algorithm 1 to solve our problem.

Finally, a brief explanation of the fact that SQA has a polynomial time complexity is shown in the following. For each decision, SQA needs to run I iterations, and the number of steps in each iteration is $STEP$. In each step, a linear time complexity $O(|\mathcal{M}|)$ is required to determine the next action while exploring and updating corresponding $\hat{Q}$ values once using backtracking. Therefore, the total time complexity of SQA for one decision is $O(I \cdot STEP \cdot |\mathcal{M}|)$, which is polynomial.

IV. NUMERICAL EXPERIMENTS

A. Experiment setup

1) Network map and UE trajectory generation: We consider a square network map with $Z \times Z$ grids, with grid lines representing roads, the length of which is set to 200m. BSs are deployed at each intersection of grid lines. We consider an area of 1600m $\times$1600m, which corresponds to 8×8 grids with 64 BSs. Each BS covers a circular area with a radius of 300m. A rectangular building with random length and width (not exceeding the road length) is randomly placed at each grid.

Due to the nature of mmWave communications, a building will block any transmissions between BSs and UEs if the building is on the Line-of-Sight (LoS) transmission path. The bandwidth B_m of all BSs is uniformly set to 10MHz, the average transmit power of all BS-UE pairs is also uniformly set as $P_{m,n} = P = 30$dBm. We also set $N_0 = -90$dBm, and $\beta = 3$.

We further assume that UEs are only distributed on the roads and only move along them. UEs move at an average speed of 15m/s towards a random possible direction, which ultimately form the predetermined trajectories. We simulate SQA and the benchmarks in four scenarios with different densities of UEs.

2) Benchmark approaches: We compare the performance of SQA with four state-of-the-art benchmarks: SNR-based handover (SBH), Rate-based handover (RBH), Learning-based handover (LBH) [14], and SMART [8]. SBH and RBH are greedy algorithms which always choose the BS with the highest SNR or instant transmission rate at a decision epoch, respectively. SMART and LBH are two recently proposed RL approaches. SMART aims to control the frequency of handovers, subject to the amount of incremental total transmission rate after a handover. LBH uses Q-learning to optimize the long-term transmission rate, assuming that a constant bandwidth is offered to every UE.

3) Experiment initialization and parameter settings: Note that an initial $\hat{Q}$ table is necessary to start the SQA. We use a greedy algorithm to initialize the $\hat{Q}$ table. In addition, we set $\epsilon = 3$, $\alpha = 0.01$, $\gamma = 1.0$, $STEP = \frac{N}{2}$ (half of the total number of UEs), and the number of iterations is set to 100.

B. Numerical results

Table I shows the simulation results. $\bar{L}$ is the long-run average transmission rate of the network, and $\bar{X}_n$ is the average time between two successive handovers for a single UE. Recall that the triggering condition for handover events is SNR, which we assumed to be influenced only by the distance between the UE and the BS. Therefore, $\bar{X}_n$ under a particular algorithm is not affected by the densities of UEs, but determined by the average speed of UEs.

(a) Very low density of UEs (BS:UE=1:8)

(b) Moderately low density of UEs (BS:UE=1:16)

(c) Moderately high density of UEs (BS:UE=1:24)
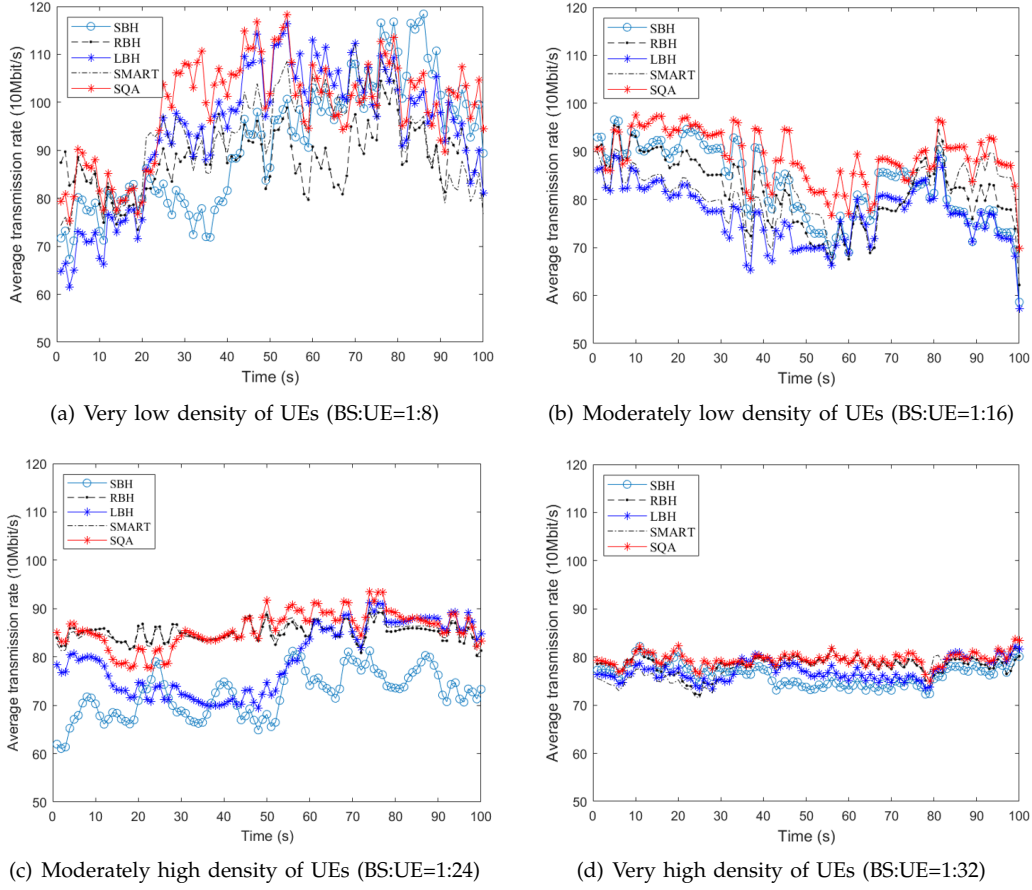
(d) Very high density of UEs (BS:UE=1:32)

Fig. 2. The comparison of instant transmission rate achieved by different algorithms in the first 100 seconds.

TABLE I
PERFORMANCE COMPARISON OF 5 ALGORITHMS IN 4 SCENARIOS
WITH DIFFERENT UE DENSITIES.

Very low density of UEs (BS:UE=64:512)					
	SBH	RBH	LBH	SMART	SQA
$\bar{L}$ (10Mbit/s)	77.60	88.09	81.64	84.67	96.26
X_n (s)	14.42	13.40	17.96	14.89	14.76
Moderately low density of UEs (BS:UE=64:1024)					
	SBH	RBH	LBH	SMART	SQA
$\bar{L}$ (10Mbit/s)	81.33	85.80	81.71	84.62	93.67
X_n (s)	14.50	13.57	17.96	14.62	14.59
Moderately high density of UEs (BS:UE=64:1536)					
	SBH	RBH	LBH	SMART	SQA
$\bar{L}$ (10Mbit/s)	80.51	85.81	82.32	83.56	87.28
X_n (s)	14.39	13.28	17.79	14.14	14.37
Very high density of UEs (BS:UE=64:2048)					
	SBH	RBH	LBH	SMART	SQA
$\bar{L}$ (10Mbit/s)	82.07	84.29	84.28	84.01	85.76
X_n (s)	14.45	14.04	17.90	14.52	14.50

In terms of $\bar{L}$, SQA outperforms four benchmarks in the four scenarios. Notably, in low density scenarios, SQA can improve over 10% throughput than other algorithms. For X_n, the performances of SQA and two RL-based benchmarks are close. We also observe that, by utilizing the future trajectory information of UEs, SQA and LBH attain significantly better $\bar{L}$ and X_n than the other three algorithms, respectively.

The greedy algorithms fail to attain the balance between $\bar{L}$ and X_n. While RBH can also achieve comparatively high $\bar{L}$, it incurs the most frequent handovers as reflected in the lowest X_n among all algorithms. On the other hand, LBH leads to the least frequent handovers, but its $\bar{L}$ is not satisfactory under scenarios with relatively low UE densities.

To demonstrate the results more intuitively, we show the instant average transmission rate achieved by different algorithms for the first 100 seconds in Fig. 2. It can be observed that when the UE density is low, SQA can keep the instant rate at a relatively high level and is remarkably superior to other algorithms. While the differences between algorithms are less significant as the UE density increases, SQA still outperforms other algorithms most times. Combining the long-run average results demonstrated in Table I, SQA is able to achieve remarkable improvements for both short and long periods of time.

V. CONCLUSION

This paper investigated the problem of optimizing the long-run average transmission rate with a feasible user association scheme in mmWave networks while

controlling the number of handovers to a reasonable level. To solve this NP-hard problem, we proposed the SQA, which has a polynomial time complexity concerning the number of BSs and UEs. Simulation results demonstrated that the SQA outperforms all benchmark algorithms, including two greedy approaches and two recently proposed RL-based algorithms, in a range of scenarios with different densities of UEs.

ACKNOWLEDGEMENT

REFERENCES

[1] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K.-K. Wong, R. Schober, and L. Hanzo, "User association in 5G networks: A survey and an outlook," *IEEE Commun. Surv. Tutor.*, vol. 18, pp. 1018–1044, 2016.

[2] H. Tabassum, M. Salehi, and E. Hossain, "Fundamentals of mobility-aware performance characterization of cellular networks: A tutorial," *IEEE Commun. Surv. Tutor.*, vol. 21, pp. 2288–2308, 2019.

[3] J. G. Andrews, T. Bai, M. N. Kulkarni, A. Alkhateeb, A. K. Gupta, and R. W. Heath, "Modeling and analyzing millimeter wave cellular systems," *IEEE Trans. Commun.*, vol. 65, pp. 403–430, 2016.

[4] M. Dong, T. Kim, J. Wu, and E. W.-M. Wong, "Millimeter-wave base station deployment using the scenario sampling approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14 013–14 018, 2020.

[5] S. Choi, J.-G. Choi, and S. Bahk, "Mobility-aware analysis of millimeter wave communication systems with blockages," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5901–5912, 2020.

[6] A. S. Cacciapuoti, "Mobility-aware user association for 5G mmWave networks," *IEEE Access*, vol. 5, pp. 21 497–21 507, 2017.

[7] R. Arshad, H. Elsawy, S. Sorour, M.-S. Alouini, and T. Y. Al-Naffouri, "Mobility-aware user association in uplink cellular networks," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2452–2455, 2017.

[8] Y. Sun, G. Feng, S. Qin, Y.-C. Liang, and T.-S. P. Yum, "The SMART handoff policy for millimeter wave heterogeneous cellular networks," *IEEE Trans. Mob. Comput.*, vol. 17, pp. 1456–1468, 2017.

[9] I. Shayea, M. Ergen, M. H. Azmi, S. A. Çolak, R. Nordin, and Y. I. Daradkeh, "Key challenges, drivers and solutions for mobility management in 5G networks: A survey," *IEEE Access*, vol. 8, pp. 172 534–172 552, 2020.

[10] S. Dixon, H. Irshad, D. M. Pankratz, and J. Bornstein, "The 2019 deloitte city mobility index," *Deloitte Insights*, vol. 18, 2019.

[11] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, 2019.

[12] M. Sana, A. De Domenico, W. Yu, Y. Lostanlen, and E. C. Strinati, "Multi-agent reinforcement learning for adaptive user association in dynamic mmWave networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6520–6534, 2020.

[13] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 124–13 138, 2020.

[14] S. Khosravi, H. Shokri-Ghadikolaei, and M. Petrova, "Learning-based handover in mobile millimeter-wave networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 2, pp. 663–674, 2020.

[15] H. Elshaer, M. N. Kulkarni, F. Boccardi, J. G. Andrews, and M. Dohler, "Downlink and uplink cell association with traditional macrocells and millimeter wave small cells," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6244–6258, 2016.

[16] 3GPP TS 36.331, "E-UTRA radio resource control (RRC); protocol specification (Release 9)," 2016.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[18] Z. Mlika, M. Goonewardena, W. Ajib, and H. Elbiaze, "User–base-station association in HetNets: Complexity and efficient algorithms," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1484–1495, 2016.

[19] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[20] T. G. Dietterich, "Hierarchical reinforcement learning with the MAXQ value function decomposition," *JAIR*, vol. 13, pp. 227–303, 2000.