

Cell Switching in HAPS-Aided Networking: How the Obscurity of Traffic Loads Affects the Decision

Berk Çiloğlu^{*,§}, Görkem Berkay Koç^{*,§}, Metin Ozturk^{*,§}, Halim Yanikomeroglu[§]

Abstract—This study aims to introduce the cell load estimation problem of cell switching approaches in cellular networks—specially presented in a high-altitude platform station (HAPS)-assisted network. The problem arises from the fact that the traffic loads of sleeping base stations for the next time slot cannot be perfectly known, but they can rather be estimated, and any estimation error could result in divergence from the optimal decision, which subsequently affects the performance of energy efficiency. The traffic loads of the sleeping base stations for the next time slot are required because the switching decisions are made proactively in the current time slot. Two different Q -learning algorithms are developed; one is full-scale, focusing solely on the performance, while the other one is lightweight and addresses the computational cost. Results confirm that the estimation error is capable of changing cell switching decisions that yields performance divergence compared to no-error scenarios. Moreover, the developed Q -learning algorithms perform well since an insignificant difference (i.e., 0.3%) is observed between them and the optimum algorithm.

Index Terms—6G, cell switching, energy efficiency, HAPS, sustainability, VHetNet

I. INTRODUCTION

Ubiquitous connectivity and *sustainability* are two imperative ingredients of the sixth generation of communication systems (6G), as they are envisioned to be key aspects in 6G networks by International Telecommunication Union (ITU) [1]. To this end, cell switching (CS), which aims to reduce the energy consumption of base stations (BSs) by putting them into a sleep mode, has been identified as a viable tool for sustainability in the literature [2]–[4]. Besides, high-altitude platform station (HAPS) can be seen as an integral part of CS in cellular networks [5], as the idle users can be offloaded to HAPS super macro BS (HAPS-SMBS) [6]; providing an extra capacity for more switching off opportunities and serving as a new BS where existing infrastructure is not sufficient. The integration of non-terrestrial networks (NTNs), through HAPS, with terrestrial networks (TNs), therefore, pave the way for ubiquitous connectivity and sustainability in 6G networks.

The literature is not ripe in terms of HAPS-assisted CS in cellular networks with only a few exceptions. For example, the study in [7], which is the earlier version of this current paper, proposed CS for HAPS-integrated TN with a simple exhaustive search (ES) algorithm. The authors in [8] investigated the case where the entire load was offloaded to the HAPS-SMBS instead of the macro cell, while in [9] sorting of small BS (SBS) payloads in ascending order was studied to transfer traffic to macro BS and HAPS-SMBS. On the

other hand, the CS literature on TNs is quite saturated with a plethora of distinctive studies. The authors in [4] studied various sleep depth levels, while the work in [10] applied the SARSA algorithm with value function approximation to reduce the energy consumption.

Although the literature has been full of CS methods taking the problem from various angles, an overlooked fact exists regardless of being TN-only or TN-NTN integration: cell load estimation problem. To the best of our knowledge, this problem has never been identified in the literature before; however, if not addressed, it has the potential of making many studies inapplicable—setting a barrier between cellular communication networks and sustainability goals. The cell load estimation problem originates from the fact that CS studies in the literature are predominantly dependent on the cell loads [4], [7]–[10]; i.e., the CS decision/optimization is performed using the traffic loads of the cells. Nevertheless, the problem is that once a BS is put on a sleep mode at a certain time slot, how can we know its traffic load at the preceding time slot in order to decide its ON/OFF status? Further, in a more deeper understanding, how can we know the traffic loads of the cells within a time slot while performing a search-based optimization? The fact is that the loads cannot be perfectly known but can be estimated with accompanying errors.

This work is the very first attempt to introduce the cell load estimation problem in the CS literature in cellular networks. Moreover, after introducing this novel problem, we build a CS optimization model by considering a HAPS-SMBS in the network (i.e., TN-NTN integration), followed by developing two different Q -learning algorithms to solve the optimization problem. We then investigate the performance of the developed algorithms with and without the estimation errors in order to observe and reveal the impacts of the cell load estimation problem on the CS performance. In this regard, the contributions of this study are given as follows:

- The cell load estimation problem is introduced for the first time in the literature with its mathematical foundations.
- We consider a HAPS-SMBS in the network for vertical offloads in an effort to provide *TN-NTN integration* and *sustainability* as envisioned in 6G networks.
- Full-scale and lightweight Q -learning algorithms are designed for accuracy and computational cost purposes, respectively.
- We investigate the results under cell load estimation errors to understand their effects on the CS performance.

A. Notation

The following notation is adopted throughout the paper: indices/subscripts i , j , and k denote users/UEs, BSs (HAPS-SMBS and SBSs), and SBSs, respectively. **Bold** characters in mathematical equations indicate a vector or matrix. Subscripts that are variables

arXiv:2405.00387v1 [cs.NI] 1 May 2024

^{*}Electrical and Electronics Engineering, Ankara Yıldırım Beyazıt University, Ankara, Türkiye. [§]Non-Terrestrial Networks (NTN) Lab, Systems and Computer Engineering, Carleton University, Ottawa, Canada.

This research has been sponsored in part by The Scientific and Technological Research Council of Türkiye (TUBITAK).

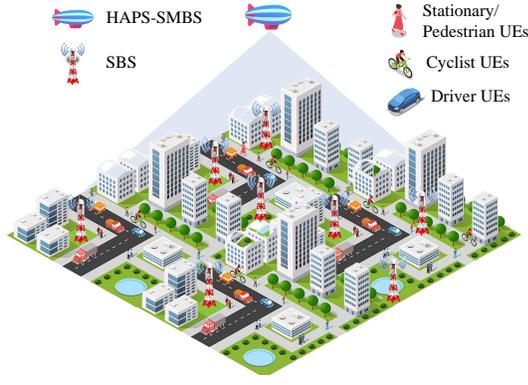


Fig. 1. A typical VHetNet model containing SBSs, HAPS-SMBS, and UEs.

are used in *italic* form while those that are non-variables and used for naming purposes only are given in straight form. Superscripts are used to differentiate the cases/scenarios for the same variable.

II. SYSTEM MODEL

A. Network Model

A vertical heterogeneous network (VHetNet) including $n \in \mathbb{N}$ SBSs, with $k=1, 2, \dots, n$ being the indices (i.e., IDs) of SBSs, and also a single HAPS-SMBS is considered as a network model. There are totally $n+1$ BSs, with $j=1, 2, \dots, (n+1)$ being the indices of BSs, are deployed consisting of n SBSs and a single HAPS-SMBS. There are also $e \in \mathbb{N}$ users/UEs, with $i=1, 2, \dots, e$ being the indices for the users/UEs, distributed around the network environment. We also introduce a user density variable, δ , which is defined by the number of users per m^2 , such that $\delta = e/A$ where A is the area (in m^2) of the network. SBSs are placed in a symmetrical manner with reference to the center of the environment and HAPS is deployed from 20 km above the ground by centralizing the environment. Due to the HAPS-SMBS's enormous footprint, it is not realistic to assume that the entire capacity of HAPS-SMBS is dedicated to our considered network, thereby we set the utilized capacity of HAPS-SMBS to a certain level: 70% of the HAPS-SMBS capacity is reserved for the network-in-question, while the rest of 30% is utilized by other networks within the footprint of HAPS-SMBS. The system model is illustrated in Fig. 2, wherein different UE mobility modes are considered: stationary, pedestrian, cyclist, and driver.

B. Propagation Model

The path loss model of HAPS-SMBS is built as described in the 3GPP report given in [11]. Since shadowing could affect path loss in various ways, it is necessary to consider the line-of-sight (LoS) and non-LoS (NLoS) situations. The path loss, L^ω , $\omega \in \{\text{LoS}, \text{NLoS}\}$, is calculated as [11]

$$L^\omega = L_b^\omega + L_g + L_s + L_e, \quad (1)$$

where L_b^ω is the basic path loss, L_g is the attenuation caused by atmospheric gasses, L_s is the loss incurred by either ionospheric or tropospheric scintillation, and L_e represents building entry loss. More details about the path loss model of HAPS-SMBS are given in [7]. For the consistency between NTN and TN path loss models, the 3GPP report given in [12] is employed for TN.

C. User-Cell Association

The user-cell association is conducted by considering the signal-to-interference-plus-noise ratio (SINR) levels, such that to associate user i to BS j , there are three certain criteria to be satisfied. Let these criteria be κ_1 , κ_2 , and κ_3 , where $\kappa_1, \kappa_2, \kappa_3 \in \{0, 1\}$ in such a way that if they are satisfied they get the value of 1 and 0 otherwise. First, a BS needs to provide the highest SINR to the UE; $\kappa_1 = \{\kappa_1 \in \{0, 1\} = 1 \mid \gamma_j > \forall \gamma_{h \neq j}, (1 \leq j, h \in \mathbb{N} \leq n+1)\}$, where γ_j denotes the instantaneous SINR of BS j at time t . Second, the BS needs to have an available capacity to accommodate UE i : $\kappa_2 = \{\kappa_2 \in \{0, 1\} = 1 \mid \Omega_j^a \geq D_i\}$, where $\Omega_j^a \in \mathbb{R}^+$ and $D_i \in \mathbb{R}^+$ represent the available capacity of BS j and the demand of UE i at time t , respectively, while $\Omega_j \in \mathbb{R}^+$ indicates the total capacity of BS j . Third, the receiver sensitivities of UEs have to be respected to assure the quality-of-service (QoS); i.e., $\kappa_3 = \{\kappa_3 \in \{0, 1\} = 1 \mid P_{\text{RX},i} \geq P_{\text{RX},\min,i}\}$, where $P_{\text{RX},i}$ and $P_{\text{RX},\min,i}$ denote the received power and receiver sensitivity of UE i at time t , respectively.

Let $U_{i,j} \in \{0, 1\}$ denote the association between UE i and BS j , such that $U_{i,j} = 1$ if UE i is associated to BS j , and $U_{i,j} = 0$ otherwise. Thus, the user-cell association is given by

$$U_{i,j} = \begin{cases} 1, & (\kappa_1=1) \wedge (\kappa_2=1) \wedge (\kappa_3=1), \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

D. Data Traffic Model

Radio resources are modeled based on resource blocks (RBs); i.e., the total capacities of BSs (SBSs or HAPS-SMBS) are divided into RBs. At each time slot, each UE generates data traffic with a certain rate, which requires a certain amount of bandwidth (i.e., RBs). Put simply, each UE demands a certain amount of RBs with a random process: $r_i = \Psi \times \mathcal{U}\{r_l, r_u\}$, where Ψ is a coefficient to shape the RB demand profile of UEs. r_l and r_u are the lower and upper bounds that a UE can demand at time t , and here they are selected as 1 and 3, respectively. Note that Ψ is obtained from a real data set provided by Telecom Italia for Milan, Italy [13].

E. Power Consumption Model

The power consumption model is adopted from the EARTH model in [14] to calculate P_B^α where $\alpha \in \{\text{S}, \text{H}\}$. Hereafter, any variable with superscript S and H indicates that the variable is for SBSs and HAPS-SMBS, respectively. The power consumption of a BS is given by

$$P_B^\alpha = \begin{cases} P_C^\alpha + \xi^\alpha \rho P_{\text{max}}^\alpha, & 0 < P_{\text{TX}}^\alpha < P_{\text{max}}^\alpha, \\ P_S^\alpha, & P_{\text{TX}}^\alpha = 0, \end{cases} \quad (3)$$

where P_C^α is constant power, ξ^α slope of the load-dependent power consumption value, $\rho = [0, 1]$ is the load factor of BSs. P_{TX}^α and P_{max}^α are the instantaneous and the maximum transmit powers of BSs, respectively. P_S^α is the power consumption when a BS is put into the sleep mode.

The total load factor of the network, ρ_T , can be defined as $\rho_T = \sum_{j=1}^{n+1} \rho_j$. Here, ρ_j can be expressed as $\rho_j = \sum_{i=1}^{e_j} r_i$, where e_j is the number of UEs that BS j accommodates. Therefore, ρ_T can be rearranged as

$$\rho_T = \sum_{j=1}^{n+1} \sum_{i=1}^{e_j} r_i, \quad (4)$$

making ρ_T a function of the number of BSs, $n+1$, the user density, δ , and the RB demand of UEs, i.e., $\rho_T = f(n, \delta, r)$, as the user density is a function of the number of users, such that $\delta = f(e)$. It is important to note here is that $e = \sum_{j=1}^{n+1} e_j$ iff $\sum_{j=1}^{n+1} \Omega_j \geq \sum_{i=1}^e r_i$, yielding the possibility of unconnected/uncovered users.

III. PROBLEM FORMULATION

This section describes the problem in a formal way by providing an optimization modeling and mathematical derivations.

Theorem 1. *The error caused by the traffic load estimation, $\varepsilon \in \mathbb{R}$, can change the policy, $\boldsymbol{\eta}$, in the optimization process, which is based on the cell loads; i.e., the objective function of the optimization problem is a function of cell loads.*

Proof. First, it is important to give the optimization model in order to reveal what kind of problem we work on. This work primarily focuses on minimizing the total power consumption of a VHetNet, consisting of several SBSs and a single HAPS-SMBS, by finding the best switching off policy. A policy $\boldsymbol{\eta}_t = [\beta_{1,t}, \beta_{2,t}, \beta_{3,t}, \dots, \beta_{n,t}]$ indicates which SBSs are ON or OFF at a given time t , where $\beta_{k,t} \in \{0,1\}$ denotes the ON/OFF status of SBS k at time t . HAPS-SMBS is assumed to be always active.

The combinatorial problem for the considered scenario, with multiple SBSs and a single HAPS-SMBS, can be modeled as [4]

$$\begin{aligned} \min_{\boldsymbol{\eta}} \quad & P_N \\ \text{s.t.} \quad & \rho \leq 1 \quad (C_1), \\ & P_{\text{TX}}^\alpha \leq P_{\text{max}}^\alpha \quad (C_2), \end{aligned} \quad (5)$$

where C_1 restricts the capacities of both SBSs and HAPS-SMBS from being violated while C_2 prevents BSs (SBS or HAPS-SMBS) from exceeding their maximum transmit power. P_N is the total power consumption of the considered VHetNet, wherein several SBSs and a single HAPS-SMBS co-exist, and can be expressed as

$$P_N = P_B^H + \sum_{k=1}^n P_{B,k}^S, \quad (6)$$

where $P_{B,k}^S$ indicates the power consumption of the k^{th} SBS.

Substituting (3) into (6) yields

$$P_N = \left[P_B^H + \sum_{k=1}^n (P_{C,k}^S + \xi_k^S \rho_k P_{\text{max},k}^S) \beta_{k,t} + P_{S,k}^S (1 - \beta_{k,t}) \right]. \quad (7)$$

Since the optimal policy, $\boldsymbol{\eta}_{\text{opt}}$, is the one that minimizes P_N by satisfying C_1 and C_2 , such that

$$\underset{\boldsymbol{\eta}}{\text{argmin}}(P_N) = \boldsymbol{\eta}_{\text{opt}}, \quad (8)$$

the decision (i.e., the policy, $\boldsymbol{\eta}$) is taken based on the total power consumption of the network, P_N , and thereby on the load factor, ρ . In other words, from (7), $P_N = f(\rho)$ and, from (8), $\boldsymbol{\eta}_{\text{opt}} = g(P_N)$, where f and g are functions. Then we can express $\boldsymbol{\eta}_{\text{opt}}$ as a function of ρ by composing f and g , such that $\boldsymbol{\eta}_{\text{opt}} = g(P_N) = g(f(\rho))$ where $(g \circ f)(x) = g(f(x))$.

Therefore, an error, ε , in the load factor would change the decision/policy; i.e., if ε is added to ρ in (7), we get an erroneous (i.e., updated with error) power consumption as

$$P'_N = \left[P_B^H + \sum_{k=1}^n (P_{C,k}^S + \xi_k^S (\rho_k + \varepsilon) P_{\text{max},k}^S) \beta_{k,t} + P_{S,k}^S (1 - \beta_{k,t}) \right], \quad (9)$$

which in turn changes the value in (8) (i.e., updates the optimal policy: $\boldsymbol{\eta}'_{\text{opt}}$) as $\underset{\boldsymbol{\eta}}{\text{argmin}}(P'_N) = \boldsymbol{\eta}'_{\text{opt}}$, where $\exists \boldsymbol{\eta}'_{\text{opt}} = \boldsymbol{\eta}_{\text{opt}}$. Therefore, when the CS optimization model includes cell loads in the decision process, the errors that occur while estimating the loads of the sleeping SBSs for the next time slot make the process erroneous, which has a potential of altering the final decision. \square

A. Cell Load Estimation Problem

The optimization problem given in (5) ensures that the total power consumption is minimized by respecting the given constraints. Since this is a combinatorial optimization problem, there are switching possibilities, in which there exists a policy (i.e., the best policy, $\boldsymbol{\eta}_{\text{opt}}$) that minimizes the total power consumption, P_N . On one hand, the optimization objective in (5) yields that the decision of selecting the best policy, $\boldsymbol{\eta}_{\text{opt}}$, actually depends on the load factors of BSs, ρ . On the other hand, since this is a combinatorial optimization problem, search or metaheuristic algorithms are employed for the solution, meaning that they have to try different combinations of active and deactive SBSs and find the best one. From there, it is worth noting that the problem is two-fold: inter-time-slot and intra-time-slot.

1) *Inter-Time-Slot Load Estimation Problem:* The question is how can we know the traffic load of an already sleeping SBS for the next time slot? In other words, determining the load factor information, ρ_k , of SBS k , which was previously scheduled to enter sleep mode in the previous time slot, is essential for deciding its status in the next time slot. However, existing literature on CS predominantly assumes perfect knowledge about traffic loads [4], [7]–[10]. The traffic loads of sleeping cells for the next time slots are needed due to the fact that CS decisions are made for one slot ahead, meaning that the decision for the next time slot, $t+1$, is made at the current time slot, t . Otherwise, at the beginning of each time slot, all the SBSs need to be switched on to get their traffic information, ρ , followed by the CS optimization, which makes the process inefficient and hard to implement.

2) *Intra-Time-Slot Load Estimation Problem:* Search and heuristic algorithms, such as ES, genetic algorithm, etc., need to search different switching options to find the best one (i.e., the least power consuming one). Nonetheless, how come the traffic loads of different switching options are acquired? The most accurate approach would be performing each option in real time, but this is quite infeasible, because when the number of SBSs is large, the search process would take long so that the switching decision cannot be taken within the time slot frame. Additionally, performing each combination in real time means that user-cell association is conducted repeatedly even within a time slot, resulting in repeated connection switching that subsequently dissatisfies the user.

In short, whether it be inter-time-slot or intra-time-slot, the fundamental fact is that the traffic loads of SBSs cannot be known perfectly; such assumptions are impractical and unrealistic. The

traffic loads should rather be estimated, which yields estimation errors, ε , in return. To this end, *the hypothesis behind this work is that estimation errors are capable of changing the decision (i.e., the switching off policy, η) in the optimization process.*

IV. METHODOLOGY

In addition to utilizing the ES algorithm as a baseline for the CS operations, we propose to employ Q -learning algorithm, which is based on the consequences of actions, in other words, penalties or rewards. Q -learning uses a matrix (i.e., Q -table) to log the results of all the action-state pairs, making it prone to create problems when the number of actions and/or states goes high. Considering this, in this work, we propose two different Q -learning designs: full-scale and lightweight. The former is more accurate as the states are made equal to all the combinations of ON/OFF status of SBSs, while the latter is simplified with redefined states—significantly reducing the number of states and, subsequently, the computational cost.

A central controller unit (CCU), located at the HAPS, comprises a processor and a database, serving various functions including acquiring cell loads from active SBSs, estimating cell loads for sleeping SBSs, and implementing Q -learning algorithms. Essentially, the CCU acts as an agent for Q -learning. When an SBS is active, it shares its traffic with the CCU via a control channel, and the traffic loads for all SBSs in the network are stored in the CCU's database. Consequently, when an SBS enters sleep mode, its traffic load can be estimated by the processor using various methodologies, such as statistical or machine learning approaches, leveraging historical data stored in the database.

A. Proposed Q -learning Designs

We consider two different Q -learning designs; namely full-scale state design (FSD) and lightweight state design (LSD). The relationship between the algorithms can be visualized as in Fig. 2.

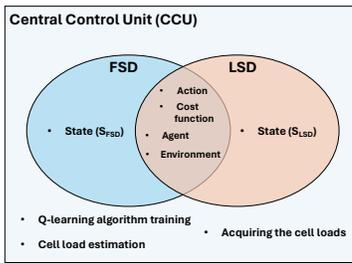


Fig. 2. The architectural commonalities and differences between the designs.

1) *Cost Function*: The cost function, which is common for both Q -learning designs, is given by

$$\Gamma(\zeta, P_N) = \zeta \times P_N + (1 - \zeta) \times \Phi, \quad (10)$$

where $\zeta = 0$ if $\sum_i \sum_j U_{i,j} < e$ and $\zeta = 1$ if $\sum_i \sum_j U_{i,j} = e$. Here, Φ is an astronomical constant penalty to avoid choosing an action that results in an unconnected UE, i.e., $\sum_i \sum_j U_{i,j} < e$.

2) *Full-Scale State Design (FSD)*: In this design, the actions, \mathbf{A}_{FSD} , and the states, \mathbf{S}_{FSD} , are equal (with dimensions of $[2^n \times n]$) and composed of all possible SBS activation/deactivation combinations. The state matrix, \mathbf{S}_{FSD} , is given by

$$\mathbf{S}_{\text{FSD}} = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \\ & & \vdots & & \\ & & \vdots & & \\ & & \vdots & & \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix}, \quad (11)$$

which is also equal to \mathbf{A}_{FSD} . The cost function is as given in (10), and the computational complexity for this design becomes $O(2^{2n})$.

3) *Lightweight State Design (LSD)*: This design is used to simplify the Q -table in FSD since it is not capable of dealing with large matrix sizes. Compared to FSD, only the set of states, \mathbf{S}_{LSD} , changes while the actions remain the same, such that $\mathbf{A}_{\text{LSD}} = \mathbf{A}_{\text{FSD}}$. The state definition becomes the number of BSs exceeding the full capacity; i.e., $s_m = \sum_{j=1}^{n+1} \Lambda_j$, where $m = 1, 2, \dots, (n+1)$ and $\Lambda_j \in \{0,1\}$ is defined as

$$\Lambda_j = \begin{cases} 1, & \sum_{i=1}^{e_j} r_i > \Omega_j, \\ 0, & \sum_{i=1}^{e_j} r_i \leq \Omega_j. \end{cases} \quad (12)$$

Here, s_m , a component of the row-vector \mathbf{S}_{LSD} that has a dimension of $[1 \times (n+1)]$, indicates the instantaneous state when the action $\mathbf{a}_{(1 \times n)}$ is taken, i.e.,

$$\mathbf{S}_{\text{LSD}} = \begin{pmatrix} s_1 \\ s_2 \\ \dots \\ s_{(n+2)} \end{pmatrix}. \quad (13)$$

The cost function is as given in (10), the computational complexity of this design is $O(2^n)$.

4) *Training Process*: In the training process, the agent (i.e., the CCU) takes an action from \mathbf{A}_{FSD} or \mathbf{A}_{LSD} , followed by calculating the cost, Γ , and updating the Q -table by $Q(s_t, \mathbf{a}_t) \leftarrow Q(s_t, \mathbf{a}_t) + \alpha[\Gamma_{t+1} + \chi \min_a Q(s_{t+1}, \mathbf{A}) - Q(s_t, \mathbf{a}_t)]$ where subscript t and $t+1$ are used to indicate *current* and *next*, respectively, and χ is the discount factor. The actions are taken based on an ϵ -greedy policy, where the agent sometimes takes random actions (following the uniform distribution) with a probability of $\epsilon = [0,1]$ (referred to as *exploration*), and sometimes more informed actions, i.e., the best one in the Q -table, (referred to as *exploitation*). Additionally, the ϵ -decaying approach is adopted, where the value of ϵ is reduced at every time slot by a certain amount in a way that $\epsilon \leftarrow \epsilon \tau$ is implemented at each time slot, where τ is the ϵ -decaying factor.

V. PERFORMANCE EVALUATION

To evaluate the impacts of load estimation problem, we use MATLAB and run two different simulation campaigns for both small- and large-scale network perspectives. In the former, $n = 4$ SBSs are utilized, while, in the latter, $n = 8$ SBSs are considered. The ES algorithm and all-active approach (A3), where all SBSs are always kept ON, are used as benchmarks. ES, FSD, and LSD algorithms are accompanied by subscripts “ $\varepsilon = 0$ ” or “ $\varepsilon > 0$ ” according to the existence of estimation error, such that if an

algorithm is run with the estimation error, then it gets the subscript of “ $\varepsilon > 0$ ”, and vice versa. Both communication- and Q -learning-related simulation parameters are given in Table I.

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Environment area (A)	1025 m \times 1025 m
Time slot number (N_{TS})	50
Time slot duration (t_d)	1 s
ε ranges	$\varepsilon_1 = [20,40]$, $\varepsilon_2 = [60,80]$, $\varepsilon_3 = [180,200]$
δ for 4SBS	{100,200,300}
δ for 8SBS	{200,400,600}
Carrier frequency (f_c)	2.5 GHz
Bandwidth (W)	50 MHz
Bandwidth per UE	200 kHz
SBSs transmit power	33 dBm
HAPS-SMBS transmit power	49 dBm [12]
SBSs antenna gain	4 dBi
HAPS-SMBS antenna gain	43.2 dBi [11]
UE antenna gain	0 dBi [12]
σ_{LoS}^2 & σ_{NLoS}^2	4 dB & 6 dB [12]
Receiver reference sensitivity	-95 dBm
SBSs constant power (P_C^{SBS})	56 W
HAPS-SMBS constant power (P_C^H)	130 W
SBSs slope of load-dependent power (ξ^{SBS})	2.6
HAPS-SMBS slope of load-dependent power (ξ^H)	4.7
SBSs maximum transmit power (P_{max}^{SBS})	6.3 W
HAPS-SMBS maximum transmit power (P_{max}^H)	20 W
Sleep mode power (P_s)	39 W
Astronomical constant penalty (ϕ)	10^9
Learning rate	0.9
Discount factor (χ) (FSD & LSD)	0.9 & 0.3
Initial ϵ value	0.8
ϵ -decaying factor (τ)	0.9
Number of iterations	20000

Energy consumption and relative difference are taken as metrics to evaluate the performance of all the considered algorithms and the impacts of load estimation errors. The energy consumption values are obtained through (3), while relative difference is defined as the percentage differences between the energy consumption values of any two CS algorithms, where the difference between the greater (always mentioned first) and lower values is divided by the greater one. The network is assessed for different UE densities, δ , which are varied in accordance with the number of UEs. In this regard, the number of UEs is chosen as 100, 200, and 300 for $n = 4$, which are duplicated when $n = 8$. Moreover, three different ε ranges, consisting of $\varepsilon_1 = [20\%, 40\%]$, $\varepsilon_2 = [60\%, 80\%]$, and $\varepsilon_3 = [180\%, 200\%]$, are considered. These ranges of ε are the percentages of the actual load factor of a BS j , ρ_j ; in the ε_1 regime, for example, the value of ε is determined randomly between 20% and 40% with a uniform distribution, followed by multiplying by ρ_j . To be more generic, the value of ε in (9) is determined by $\varepsilon = \rho_j \varepsilon_v$ where $v \in \{1, 2, 3\}$.

Fig. 3 shows the energy consumption results with respect to δ for three different ε values when $n = 4$. The first takeaway from these results is that as ε increases, the differences between the values in $ES_{\varepsilon=0}$ and the other algorithms become higher. This is because the error is unidirectional (i.e., it always adds up on the load: an assumption) and thus when it goes higher the energy consumption rises as well due to the relation in (9). Second, both $FSD_{\varepsilon>0}$ and $LSD_{\varepsilon>0}$ yield almost the same results with the $ES_{\varepsilon>0}$; there is a maximum of 0.3% relative difference between both Q -learning designs and $ES_{\varepsilon>0}$ for all ε and δ values. As $FSD_{\varepsilon>0}$ and $LSD_{\varepsilon>0}$ give almost the same results, they seem overlapped in Fig. 3. This showcases how accurate our Q -learning designs work, considering that the ES algorithm is optimum. Comparing the Q -learning designs with each other, the simulation results demonstrate

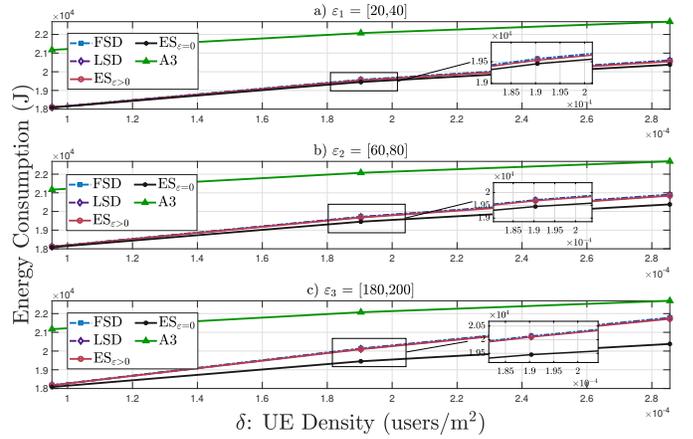


Fig. 3. Energy consumption for $n = 4$ SBSs with three different ε values.

that $LSD_{\varepsilon>0}$ gives its closest performance to $FSD_{\varepsilon>0}$ for ε_1 and $\delta = 9.5 \times 10^{-5}$ values, and the relative difference between $LSD_{\varepsilon>0}$ and $FSD_{\varepsilon>0}$ diverges up to 0.16% as ε and δ rise. However, this is still a good result given that we carry the LSD algorithm with less computational cost and it manages to reach almost the same result as FSD. When A3 and $ES_{\varepsilon=0}$ are compared, the maximum relative difference is obtained as 14.6% for $\delta = 9.5 \times 10^{-5}$. Moreover, the relative difference between A3 and $ES_{\varepsilon>0}$ is found to be 14.54% for ε_1 , which becomes 14.18% when ε_3 takes place for the same δ value. As δ increases, the relative differences between A3 and both $ES_{\varepsilon=0}$ and $ES_{\varepsilon>0}$ cases keep decreasing down to 10.15% and 4.24%, respectively. The inference behind this is that as the network becomes denser, all the CS algorithms behave closely with A3 because it is not wise to switch off an SBS due to the high demand.

Fig. 4 demonstrates the energy consumption results with respect to δ values when $n = 8$. Since the network is larger in size now, we double the number of UEs, therefore δ values are multiplied as well, as the size of the environment remains the same. The results keep a similar trend with the outcomes in Fig. 3. Another similarity is that LSD performs very close to $ES_{\varepsilon>0}$ when $n = 8$. The results illustrate that the maximum and minimum relative differences between $LSD_{\varepsilon>0}$ and $ES_{\varepsilon>0}$ are found as 3.0% and 0.11%, respectively, among all ε and δ values. These confirm that the LSD algorithm works well even when the network is denser. Note that there is no FSD algorithm included in this set of simulations due to the implementation challenges. When A3 and $ES_{\varepsilon=0}$ are put into the consideration, the maximum relative difference is obtained as 18.09% for the $\delta = 19.03 \times 10^{-5}$. The relative difference between A3 and $ES_{\varepsilon>0}$ is found as 18.04% in the ε_1 regime, which declines as low as 17.76% when ε_3 is in place for the same δ . As δ increases, the relative differences between A3 and for both $ES_{\varepsilon>0}$ and $ES_{\varepsilon=0}$ drop to 12.28% and 8.8%, respectively.

From Theorem 1, we understand that the estimation error, ε , has the potential to alter the CS policy, η . This change, in turn, influences network traffic performance and user data rates, as users may become associated with sub-optimal BSs when the CS policy differs from the optimal one, denoted as η_{opt} , leading

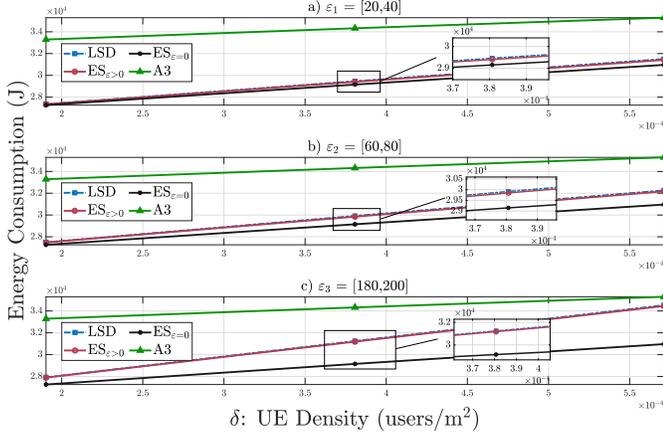


Fig. 4. Energy consumption for $n = 8$ SBSs with three different ε values.

to $\eta \neq \eta_{\text{opt}}$. This deviation from the optimal policy can result in both under-connection, where fewer users are connected compared to the optimal policy, and over-connection, where more users are connected. Such variations have a direct impact on network traffic performance. Furthermore, as discussed in [15], an increase in ε leads to a reduction in user data rates. This reduction occurs because users may be connected to BSs that do not provide the expected SINR performance, as dictated by the optimal policy η_{opt} . As a result, users experience lower data rates, as they typically select BSs with the best SINR performance.

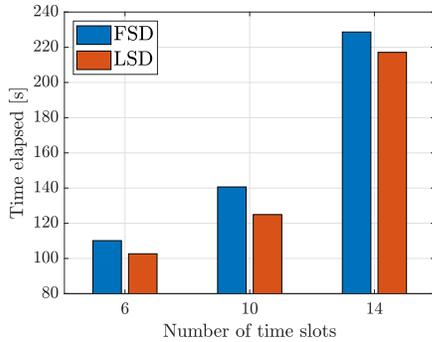


Fig. 5. Elapsed time comparison between FSD and LSD for different number of time slots when $n = 4$. Results are averages of 10 runs (Monte Carlo).

We conducted an empirical test on the complexities of both designs, as illustrated in Fig. 5. The simulations were performed using a computer with the following specifications: Processor: 12th Gen Intel(R) Core(TM) i5-1230U @ 1.00 GHz; RAM: 8GB (7.7 GB usable); System Type: 64-bit operating system, x64-based processor. The results indicate that FSD results in higher complexity (i.e., elapsed time) compared to the LSD. However, the difference is not as significant as the theoretical computational complexity discussed in Section IV-A using big- O notation. This observation is attributed to the fact that LSD and FSD differ primarily in the state design (i.e., dynamic part), while they share several common components such as actions and user-cell associations. As a result, the static part dominates the overall complexity, making the difference in

the dynamic part less significant. Furthermore, additional analyses reveal that the difference in complexity increases by approximately 21% when the number of SBSs, n , increases from 4 to 8 due to the larger size of the Q -matrix.

VI. CONCLUSION

The impacts of cell load estimation errors for CS operations in cellular networks were introduced for the first time in this study. Moreover, two different Q -learning algorithms were developed, and in order to test the algorithms and observe the impacts of cell load estimation errors a VHetNet was considered, which includes the promising HAPS technology. The results primarily confirmed the hypothesis behind the paper that the cell load estimation error is able to change the decision taken by optimization algorithms. Further, both Q -learning algorithms performed reasonably well compared to the ES algorithm. In future, we plan to develop solutions to estimate cell loads more accurately by means of artificial intelligence.

REFERENCES

- [1] "The ITU-R Framework for IMT-2030," ITU-R Working Party 5D, International Telecommunication Union (ITU), Tech. Rep., July 2023. [Online]. Available: <https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2030/Pages/default.aspx>
- [2] F. Ahmed, M. Naeem, W. Ejaz, M. Iqbal, A. Anpalagan, and M. Haneef, "Energy cooperation with sleep mechanism in renewable energy assisted cellular hetnets," *Wireless Personal Communications*, vol. 116, pp. 105–124, 2021.
- [3] X. Tan, K. Xiong, B. Gao, P. Fan, and K. B. Letaief, "Energy-efficient base station switching-off with guaranteed cooperative profit gain of mobile network operators," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 3, pp. 1250–1266, 2023.
- [4] A. E. Amine, J.-P. Chaiban, H. A. H. Hassan, P. Dini, L. Nuaymi, and R. Achkar, "Energy optimization with multi-sleeping control in 5G heterogeneous networks using reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4310–4322, 2022.
- [5] D. Renga and M. Meo, "Can high altitude platform stations make 6G sustainable?" *IEEE Communications Magazine*, vol. 60, no. 9, pp. 75–80, 2022.
- [6] M. S. Alam, G. K. Kurt, H. Yanikomeroglu, P. Zhu, and N. D. Dao, "High altitude platform station based super macro base station constellations," *IEEE Communications Magazine*, vol. 59, no. 1, pp. 103–109, 2021.
- [7] G. B. Koç, B. Çiloğlu, M. Öztürk, and H. Yanikomeroglu, "HAPS-enabled sustainability provision in cellular networks through cell-switching," in *2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2023, pp. 294–299.
- [8] T. Song, D. Lopez, M. Meo, N. Piovesan, and D. Renga, "High altitude platform stations: the new network energy efficiency enabler in the 6G era," *arXiv preprint arXiv:2307.00969*, 2023.
- [9] M. Salamtmoghadas, A. Mehrabian, and H. Yanikomeroglu, "Energy sustainability in dense radio access networks via high altitude platform stations," *IEEE Networking Letters (Early Access)*, 2023.
- [10] M. Ozturk, A. I. Abubakar, J. P. B. Nadas, R. N. B. Rais, S. Hussain, and M. A. Imran, "Energy optimization in ultra-dense radio access networks via traffic-aware cell switching," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 832–845, 2021.
- [11] "Study on New Radio (NR) to support non-terrestrial networks," 3rd Generation Partnership Project (3GPP), TR 38.811, Sep. 2020.
- [12] "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), TS 38.901, May 2017.
- [13] "Telecommunications - SMS, Call, Internet - MI," Telecom Italia, 2015. [Online]. Available: <https://doi.org/10.7910/DVN/EGZHFV>
- [14] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume, and A. Fehske, "How much energy is needed to run a wireless network?" *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40–49, Oct. 2011.
- [15] G. B. Koç, B. Çiloğlu, M. Öztürk, and H. Yanikomeroglu, "A lightweight machine learning approach for delay-aware cell-switching in 6G haps networks," to appear in *IEEE International Conference on Communications (ICC) Workshops* (arXiv preprint arXiv:2402.13096), 2024.