D-RAN: A DRL-based Demand-Driven Elastic User-Centric RAN Optimization for 6G & Beyond

Shahrukh Khan Kasi, *Student Member, IEEE,* Umair Sajid Hashmi, *Member, IEEE*, Sabit Ekin, *Senior Member, IEEE*, Adnan Abu-Dayya, *Senior Member, IEEE*, and Ali Imran, *Senior Member, IEEE*

***Abstract*—With highly heterogeneous application requirements, 6G and beyond cellular networks are expected to be demand-driven, elastic, user-centric, and capable of supporting multiple services. A redesign of the one-size-fits-all cellular architecture is needed to support heterogeneous application needs. While several recent works have proposed user-centric cloud radio access network (UCRAN) architectures, these works do not consider the heterogeneity of application requirements or the mobility of users. Even though significant gains in performance have been reported, the inherent rigidity of these methods limits their ability to meet the quality of service (QoS) expected from future cellular networks. This paper addresses this need by proposing an intelligent, demand-driven, elastic UCRAN architecture capable of providing services to a diverse set of use cases including augmented/virtual reality, high-speed rails, industrial robots, E-health, and more applications. The proposed framework leverages deep reinforcement learning to adjust the size of a user-centered virtual cell based on each application's heterogeneous requirements. Furthermore, the proposed architecture is adaptable to varying user demands and mobility while performing multi-objective optimization of key network performance indicators (KPIs). Finally, numerical results are presented to validate the convergence, adaptability, and performance of the proposed approach against meta-heuristics and brute-force methods.**

***Index Terms*—User-centric, elastic architecture, demand-driven, deep reinforcement learning, spectral efficiency, energy efficiency, throughput.**

I. Introduction

A. Background

A key feature of 6G and beyond networks will be ultra-dense networks offering seamless coverage, very high throughput, and ultra-low latency. Network operators are exploring ultra-dense networks to meet the ever-growing demand for throughput and latency envisioned for 6G and beyond users. While researchers in both academia and industry agree that network densification will enhance the coverage and capacity of current cellular networks, it has its own complications [1]. By densifying the network, the average distance between users and the interferring base stations reduces. This causes a shift in pathloss exponent leading to a scenario where increase in the interference from neighboring base stations overshadows benefits of the decreased average distance from serving base stations. Earlier models that relied on single slope pathloss models did not capture this phenomenon and led to the belief that the distribution of signal-to-interference-noise ratio (SINR) is independent of the base stations density. However, use of more realistic multi-slope pathloss models in recent works has led to the debunking of this myth, while proving that dense networks are inherently interference-limited [2].

Further, 6G communications are envisioned to cater to a wide range of user services with assorted throughput and latency requirements [3]. In order to meet this requirement, there is a need for an elastic architecture that can tailor to the needs of each service, as opposed to traditional one-size-fits-all architecture. This, along with the interference-limited nature of dense networks, has prompted a shift to a user-centric network paradigm from traditional networks. [4]–[6].

UCRAN's ability to abate inter-cell interference and reduce deployment/operational costs makes it the ideal architecture for supporting user-centric services in dense cellular networks [7]. A typical UCRAN consists of a tier of low-density large coverage control base station (CBS) underlaid by a tier of high-density intermediate coverage switchable data base stations (DBS). UCRAN introduces a new degree of freedom that is elastic in nature, referred to as Service Zone or S-zone in this paper. S-zone is defined as the size of the user-centric virtual cell centered around scheduled user equipment(s) (UEs). In each transmission time interval (TTI), CBS activates the best DBS constituting a S-zone centered on the scheduled UE while ensuring no overlap among S-zones. With this concept, the macro-diversity gain is easily achieved through the activation of the best DBS for a scheduled UE.

B. Related Work

In recent works, the impact of S-zone size in a UCRAN is investigated using analytical models for both sub 6 Gigahertz and millimeter frequency bands [5], [7]–[11]. The network design in these works considers creating a non-overlapping virtual cell (S-zone) around scheduled users that are scheduled based on their priorities. The non-overlapping user-centric cells and macro-diversity technique allows S-zone size to be employed as a control parameter that can be optimized based on the desired KPIs.

For instance, Hashmi et. al. [5] using a statistical framework showed that there exists an optimal user-centric virtual cell size at which both the area spectral efficiency and energy efficiency can be maximized in UCRAN. The authors also noted that this user-centric virtual cell size depends on both DBS and user

Shahrukh Khan Kasi, and Ali Imran are with the AI4Networks Research Center, School of Electrical & Computer Engineering, University of Oklahoma, Tulsa, OK, USA.

Umair Sajid Hashmi is with the School of Electrical Engineering & Computer Science, National University of Sciences & Technology, PK.

Sabit Ekin is with the School of Electrical & Computer Engineering, Oklahoma State University, Stillwater, OK, USA.

Adnan Abu-Dayya is with the Department of Electrical Engineering, Qatar University, Doha, Qatar.

density variations, thus requiring adaptation with variations in these parameters. Hashmi et. al. [7] considers UCRAN based on Stienen cells to characterize the SINR distribution, area spectral efficiency, and energy efficiency as functions of user-centric virtual cell coefficient, user scheduling probability, and DBS density. They analyzed UCRAN in comparison with non user-centric architectures, demonstrating that it not only provides better SINR, but also can optimize area spectral efficiency and energy efficiency by adjusting the design parameters. In an analytical study, we studied the interaction between spectral and energy efficiencies in a coordinated multipoint-enabled UCRAN architecture as the size of the UCRAN's virtual cell and the density of its DBSs is changed [9]. Humadi et. al. [8] have proposed a user-centric model for combining base stations for millimeter-wave networks and used stochastic geometry to determine the coverage probability and optimal area spectral efficiency performance. They propose a framework for optimizing the clustering parameter, leading to increase in area spectral efficiency.

Although the existing literature on UCRAN provides some useful information, it has two shortcomings. First, it deals with static S-zone size for all UEs with the assumption that all UEs will have similar throughput and latency requirements which is not a practical assumption. Second, although the analytical models in above studies are highly detailed, they lack the interaction of controlling parameters (S-zone) with the spatiotemporal changes in the wireless network such as dynamic user application demands and mobility.

C. Motivation and Contributions

With user-centric services being considered as an essential feature of future cellular communications, 6G in particular, an elastic and demand-driven UCRAN is needed in which UEs with various throughput and latency requirements are assigned different S-zones. In this study, we present such an elastic and demand-driven UCRAN model, detailed in Section II. We formulate a multi-objective optimization problem to maximize important KPIs such as area spectral efficiency, network energy efficiency, user service rate, and throughput satisfaction. The S-zone size serves as a control parameter to form a Pareto-optimal trade-off among these KPIs.

The core research objective of this work is to develop a solution that can dynamically solve this multi-objective optimization problem in UCRAN to achieve a Pareto-optimal solution in real-time based on changes in the varying application demands and user mobility. Inspired by our earlier work on utilizing wireless network telemetric big data for enabling zero touch optimization in future wireless networks [12], we propose a deep reinforcement learning (DRL)-based framework to solve this problem. This framework is hereafter referred to as D-RAN: a deep reinforcement learning-based user-centric RAN optimization framework under dynamic user application demands and network conditions. D-RAN uses the massive amount of control, signaling, and contextual data in UCRAN network to update network parameters dynamically to optimize the KPIs of interest in real-time.

Driven by the above motivations, this paper studies the deep reinforcement learning approach owing to its ability to adapt to dynamic environments to determine the optimal S-zone size for each QoS category intelligently so that network KPIs such as area spectral efficiency, energy efficiency are maximized as well as throughput, and latency requirements of each QoS category are met. To the best of our knowledge, this is the first work to consider the allocation of dynamic S-zones to different QoS categories with eclectic throughput and latency demands. Specifically, the contributions of this paper are summarized as follows.

- An architecture for demand-driven elastic user-centric communication is proposed with the aim of providing on-demand services to a diverse set of user applications ranging from augmented/virtual reality to industrial robots to E-health applications, and more. The proposed architecture allows the elastic user-centered S-zone to be malleable to specific QoS category requirements.
- Considering the heterogeneous user requirements in future cellular communications, a multi-objective problem is formulated to optimize KPIs such as area spectral efficiency, energy efficiency, user service rate, and throughput satisfaction as a function of S-zone size for respective QoS categories. Given the stringent requirement of very high throughput and ultra-low latency, the multi-objective problem is geared towards meeting users' throughput and latency requirements while also maximizing the area spectral efficiency and network energy efficiency.
- Given the non-stationarity of user application demands and mobility, we propose a deep reinforcement learning framework to accurately learn the mapping of environment state and action instilling intelligence in the demand-driven elastic user-centric architecture. The proposed intelligent deep reinforcement learning framework for UCRAN networks, named D-RAN, dynamically allocates S-zones to users such that a Pareto-optimal front is found for the formulated multi-objective function.
- We evaluate the convergence, efficacy, and adaptability of D-RAN to the non-stationary environment of the proposed approach through numerical results. We also compare D-RAN's performance against brute-force and state-of-the-art metaheuristics such as simulated annealing. The simulation results show that D-RAN can achieve a gain of up to 45% in the network-wide utility compared to an simulated annealing-based solution. This paper has the potential to change network mode from rigid cell-centric to elastic user-centric through the use of an intelligent module (D-RAN) that allows optimization of S-zones in real-time, resulting in enhanced user experience, greater system capacity, and improved energy savings.

The remainder of the paper is organized as follows. The system model is discussed in Section II. A multi-objective optimization problem as a function of S-zone size is formulated in Section III. A brief summary of deep reinforcement learning and simulated annealing algorithms are presented in Section IV. The details of the proposed approach and the results of the numerical analysis are presented in Section V and Section VI, respectively. Finally, the paper is concluded in Section VII.

II. System Model

This section presents the UCRAN architecture, S-zones scheduling algorithm, network model, and channel model.

A. UCRAN Architecture

Fig. 1 provides a graphical illustration of a UCRAN network with virtual user-centric cell boundaries for UEs belonging to different QoS categories. These categories are classified according to the UEs' latency and throughput requirements as illustrated in Fig. 1. The DBSs are connected to the pool of base band units (BBUs) via flexible back haul (an optical fiber network) [13], [14]. Most of the signal processing at baseband level is delegated to the BBUs.

The virtual user-centric cell (S-zone) formation around a UE effectively: i) enables interference protection by inducing a guard-zone between scheduled UE and interfering DBSs (i.e., UE user-centric cells do not overlap) paving the way for increase in the system-wide SINR, throughput, and spectral efficiency; ii) enables energy saving by selectively activating a DBS when required to serve a scheduled UE, hence making the network energy efficient; iii) enables provision of seamless service experience to UEs (belonging to a panoply of traffic types) by providing demand-based coverage.

A critical design parameter in UCRAN is the size of S-zone which is defined by the radius of circular disk around the UE. In the proposed model, the DBSs falling within the S-zone of a UE are only allowed to associate with that UE in a given TTI. Increasing the S-zone size ensures (i) larger distances between a UE and interfering DBSs resulting in high link-level SINR (hence, link-level high throughput and spectral efficiency); (ii) yields high macro diversity gain through selection among the larger number of DBSs in the S-zone and (iii) offers high energy efficiency as large S-zones keep more DBSs deactivated as compared to small S-zones. However, larger S-zones also yield low user scheduling ratio and low spectrum reuse resulting in negative impact on the system-level capacity. Given these insights, the S-zone size serves as a controlling parameter that yields an ideal tradeoff between area spectral efficiency, energy efficiency, and other system-level KPIs.

In UCRAN, a scheduled user in each TTI is allocated the full bandwidth of the system for two reasons: i) to make the system capable of providing maximum throughput to a user that the total system bandwidth allows; ii) to keep the radio resource scheduling at DBS simple and thus keep DBS cost and energy consumption low. The spectrum waste is avoided by managing the temporal scheduling where a user needing a lower throughput is scheduled after larger number of TTIs. The temporal gap in TTIs after which a user is scheduled is inversely proportional to user bandwidth/throughput requirement.

Besides, to make the spectrum allocation more efficient, there is a need to intelligently allocate both physical resource blocks and S-zone size to scheduled users according to their needs. Since the D-RAN framework is proposed mainly to establish that the S-zone size of multiple QoS categories (with varied QoS demands) can be intelligently controlled to optimize the desired KPIs, the joint optimization of S-zone size and physical resource blocks will be addressed in future research. It is also important to mention that the intelligent allocation of physical resource blocks in 5G cellular systems has already been proposed in several publications [15], [16].

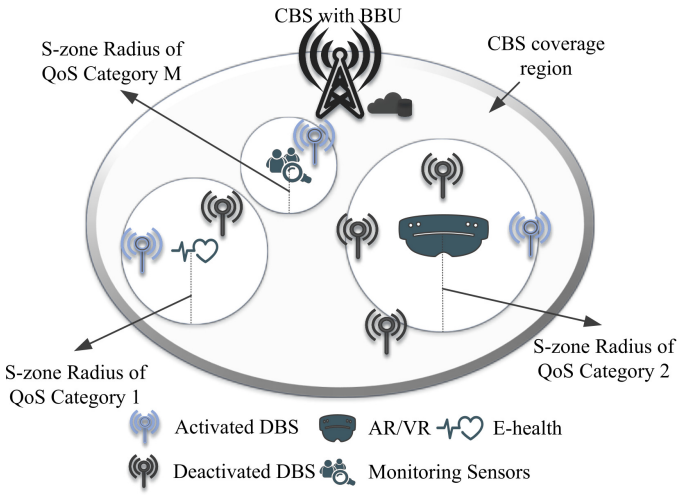


Fig. 1: Dynamic S-zone UCRAN architecture with M different S-zone region of radius R_c for scheduled UE's.

B. UE Scheduling Algorithm

In this work, we propose a scheduling mechanism to meet the heterogeneous latency requirement of UEs in UCRAN. Latency requirements of UEs are drawn from a uniform distribution and rounded off to specified bins of latency requirements corresponding to the QoS categories. Each UE x is marked with $p_x^{latency} \sim U(a,b)$ by the BBU where a and b are measured in milliseconds (ms) and are determined by the minimum and maximum latency of the considered QoS categories. The lower the value of mark $p_x^{latency} \sim U(a,b)$, the higher will be the scheduling priority.

Algorithm 1: UE Scheduling Algorithm

```
Initialize the set of UEs and the DBS(s) ;
Assign priorities to UEs based on their latency
  requirements ;
Sort UEs in the descending order according to their
  priorities ;
for each UE in the sorted list do
    if DBS available in S-zone region of UE and UE is
      not overlapping with other scheduled UEs then
        Schedule UE
```

The BBU based on these scheduling priorities schedules a UE x if and if only the scheduling priority of UE x is highest in the neighborhood which is characterized by the S-zone size R_c for a specific QoS category. This means that within a circle of radius R_c centered at UE x, no other UE has a higher priority than UE. For example, the scheduled UEs shown in Fig. 2 have a lower latency requirement than any other UE in the S-zone of the respective QoS category. Note that larger the S-zone size of QoS categories, lesser the number of UEs will be scheduled with non-overlapping S-zones.

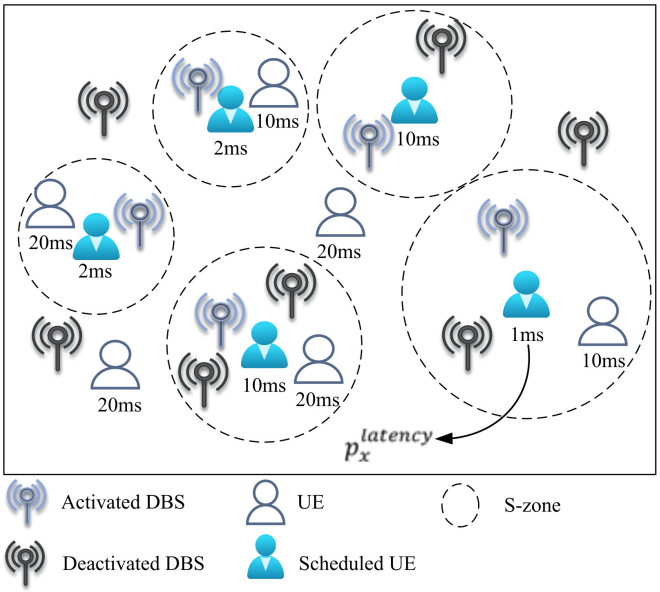


Fig. 2: Graphical illustration of UEs scheduling with varied latency requirements.

Once the UE is scheduled, a single DBS providing the highest channel gain within the S-zone of the respective UE is activated by the BBU to serve the UE. It is important that the DBSs are deployed densely, so at least one DBS is available within an S-zone to provide coverage to a scheduled UE and thus avoid coverage holes in areas where no DBSs are available within the user-centric circular disk. The scheduled UEs with no DBS within their S-Zones are served by CBS.

C. Network Model

A downlink of a two-tier ultra-dense network is considered consisting of a CBS and DBSs operating on sub 6 GHz frequencies. The DBSs and UEs are randomly distributed following two independent and homogeneous Poisson point processes Π_{DBS} and Π_{UE} with intensities λ_{DBS} and λ_{UE} respectively. The location of each UE acts as a centering point for the user-centric virtual cell (S-zone) which bounds the UE to be associated with DBS only within the S-zone region. This implies that each DBS can at most serve a single UE. This work defines the S-zone as a disk of radius R_c, where $c \in C$ is a QoS category present in the network model. The network model in Fig. 1 for example, includes three QoS categories: augmented/virtual reality, E-health, and monitoring sensors.

D. Channel Model

The communication channel between an arbitrary user $x \in \Pi_{UE}$ and activated DBS $i \in \Pi'_{DBS}$ is modeled to experience both large-scale and small-scale fading given by hl^{-PLE}, where h is an exponential random distribution with unit mean, l_{xi} represents the propagation distance between x and i, $PLE < 2$ is the pathloss exponent, and Π'_{DBS} is the Poisson point process of activated DBSs. UE and DBS are equipped with a single antenna and the transmission power of DBS is assumed to be equal. Each scheduled user is served by a DBS providing the highest channel gain within an S-zone of radius R_c whose SINR (Γ_x) is given as:

$$\Gamma_x = \frac{h_{xi} l_{xi}^{-PLE}}{\sum\limits_{j \in \Pi'_{DBS}} h_{xj} l_{xj}^{-PLE} + n_o}, \quad (1)$$

where $i \neq j$ and n_o denotes the additive white Gaussian noise.

III. Problem Formulation

This section characterizes the KPIs, followed by the formulation of a multi-objective optimization problem.

A. Characterizing Key Performance Indicators

This work measures system performance in terms of area spectral efficiency, network energy efficiency, user service rate, and throughput satisfaction as the desired set of KPIs. We selected these KPIs to reflect that the objective is to meet throughput and latency requirements while maximizing area spectral efficiency and network energy efficiency.

1) Area Spectral Efficiency

The area spectral efficiency refers to the amount of information that can be transmitted from a DBS per unit bandwidth channel per unit area to a UE, which can be defined as follows for each QoS category c:

$$\mathsf{A}_c = \frac{\sum\limits_{x \in N_c} \log_2(1+\Gamma_x)}{\text{Å}}, \quad (2)$$

where N_c is the set of UEs belonging to QoS category c, and Å is the target area considered in the simulations model.

There is a strong relationship between the QoS category's S-zone size and area spectral efficiency [5], [9]. Intuitively, increasing the S-zone size decreases the scheduling ratio of UEs. Nevertheless, decreasing the S-zone size increases the SINR (due to the higher number of neighboring interfering DBSs). There is, therefore, an optimal size for S-zones that balances these two opposing effects to maximize the attainable area spectral efficiency. To optimize the area spectral efficiency, intelligent real-time optimization is needed to calibrate the S-zone size of multiple QoS categories simultaneously.

2) Energy Efficiency

According to [5], [17], [18], the network-wide energy efficiency is defined as the ratio of area spectral efficiency and total power consumed for all scheduled UE's. The power consumption model in this paper is inspired by project Earth [19], in that it represents the power consumption of CBS and DBSs as a linear combination of fixed power and load-dependent power consumption components. Since energy efficiency is measured network-wide, these power consumption values are summed for all scheduled users. The total power consumption can be mathematically calculated as follows:

$$P = \lambda_{DBS} P_f + \lambda'_{DBS} \Delta_{DBS} P_{DBS} + \lambda'_{UE} (\Delta_{UE} P_{UE} + P_{disc}), \quad (3)$$

where λ_{DBS} is the density of all deployed DBSs, λ'_{DBS} is the density of activated DBSs, λ'_{UE} is the density of scheduled UEs, P_f is the fixed DBS power consumption required for DBS to operate in listening mode, P_{DBS} is the DBS transmission power, Δ_{DBS} is the radio frequency component power at DBS, P_{UE} is the UE transmission power, Δ_{UE} is the radio frequency component power at UE, P_{disc} is the power required

TABLE I: Power consumption parameters.

Symbol	Parameter Name	Parameter Value
P_f	DBS fixed power consumption	1.932 W
P_{DBS}	DBS transmit power	10 W
Δ_{DBS}	Radio frequency component's power consumption at DBS	23.22 W
P_{UE}	UE transmit power	1 W
Δ_{UE}	Radio frequency component's power consumption at UE	4 W
P_{disc}	UE cell discovery circuit power consumption	4.3 W

at UE for discovery of the DBS with the highest channel gain. The typical values of these variable are summarized in Table I [7]. The energy efficiency therefore can be given as:

$$\mathsf{E} = \frac{\mathring{\mathrm{A}} \times \sum\limits_{c \in C} \mathsf{A}_c}{P}. \tag{4}$$

In a cellular DBS, radio frequency components and data transmission account for the majority of total power consumption [20]. DBSs can save significant amounts of energy when they are dynamically activated, particularly in dense deployments. The direct relationship between energy efficiency and area spectral efficiency mandates that the S-zone size of QoS categories will also influence network energy efficiency. Intuitively, increasing the S-zone size decreases the number of activated DBSs (decreasing the average power consumption). The contrasting trends of area spectral efficiency and power consumption raise an important design question: what S-zone size should be selected for QoS categories to optimize network-wide energy efficiency.

3) UE Service Rate

The UEs' heterogeneous latency requirements necessitate scheduling more UEs within each TTI while meeting UE quality of experience requirements. The mean UE service rate (user service rate) for any QoS category c can be calculated as:

$$\mathsf{U}_c = \frac{\lambda_{UE_c}^{service}}{\lambda_{UE_c}}, \tag{5}$$

where λ_{UE_c} is the density of all UEs belonging to QoS category c and $\lambda_{UE_c}^{service}$ is the density of UEs belonging to QoS category c whose minimum throughput requirement is met.

The S-zone size of QoS categories influences the user service rate in two different ways. A decrease in the S-zone size leads to the scheduling of more users. However, decreasing the S-zone size also increases the average distance between UE and DBS, thus, affecting the average SINR. Due to these contrasting results with the change in S-zone size, we anticipate that optimizing user service rate will require intelligent optimization of S-zone sizes of QoS categories.

4) Throughput Satisfaction

There can be a wide variety of throughput requirements for UEs belonging to different QoS categories. Operators must satisfy the minimum throughput requirements of each QoS category as part of their objective. Moreover, network operators must ensure that they are utilizing their resources efficiently by avoiding scenarios in which excess throughput is allocated to a few UEs (or categories of UEs) while other UEs' minimum requirements are not met. For this reason, this work uses the difference between required and obtained throughput, a metric we define as throughput satisfaction (throughput satisfaction), to measure system performance. Throughput satisfaction for a specific QoS category c is given as:

$$\mathsf{T}_c = \prod_{x \in N_c} \left| tp_x^{\star} - tp_x^{\Diamond} \right|^{|N_c|}, \tag{6}$$

where $tp_x^{\star}$ and $tp_x^{\Diamond}$ are the obtained and required throughput for an arbitrary UE x respectively. The required throughput values for UEs are drawn from a uniform distribution and rounded off to specified bins of throughput requirement of QoS categories. While the obtained throughput values are obtained by mapping the SINR values of UEs to its physical layer throughput given in [21].

Intuitively, the increase in S-zone size of QoS category is expected to improve the average SINR (and throughput obtained) at the UE. However, the mere increase in throughput of a few users is not the desired behavior. Instead, the S-zone size should be adjusted such that the throughput achieved at UEs belonging to a QoS category float near the throughput requirement of that specific QoS category. This entails that the S-zone size of QoS categories should be carefully calibrated to ensure satisfaction is achieved throughput across all QoS categories.

5) Multi-objective Optimization Problem Formulation

Hitherto, the above definition of KPIs demonstrate the need for optimizing S-zone size of QoS categories to maximize area spectral efficiency, energy efficiency, UE service rate and throughput satisfaction individually. The challenge from a network operator's perspective is that all these KPIs should be optimized simultaneously, leading to a Pareto-optimal tradeoff between them. To account for this tradeoff, this study defines the multi-objective optimization problem as follows:

$$\max_{R_c} \quad \frac{\left(\sum\limits_{c \in C} \mathsf{A}'_c\right)^{\alpha} \left(\sum\limits_{c \in C} \mathsf{U}'_c\right)^{\beta} \left(\mathsf{E}'\right)^{1-\alpha-\beta}}{\sum\limits_{c \in C} \mathsf{T}'_c} \tag{7}$$
$$\text{s.t.} \quad R_{min} \leq R_c \leq R_{max},$$

where $0 \leq \alpha, \beta \leq 1$, $\alpha + \beta \leq 1$, A'_c is area spectral efficiency normalized between $[0, 1]$, E' is energy efficiency normalized between $[0, 1]$, U'_c is UE service rate normalized between $[0, 1]$, T'_c is throughout satisfaction normalized between $[1, 2]$, R_{min} and R_{max} are the minimum and maximum allowable size for S-zone of QoS categories. To bring it to the reader's attention, throughput satisfaction is included in denominator to ensure that the increase in the difference between required and

obtained throughput of UEs reduces the utility of the solution.

The rationale behind the proposed objective function formulation is to optimize holistic system-level performance by combining network operators' four most important and common KPIs of interest. However, these KPIs have different scales/units. This issue makes combining the multiple KPIs in a single objective function far from a straightforward problem. In this work, we address this problem by normalizing each KPI value with its minimum and maximum value. These minimum and maximum KPI values are determined through pseudo brute force method. The pseudo brute force method sweeps the solution space (with a pre-defined step size) in numerous independent runs. Given the step sizes are large enough to explore the possible extrema in the search space within an affordable computational effort, this pseudo brute force method gives values of KPIs that can be taken as approximation of minimum and maximum values for the normalization purposes. This way of approximating the true Pareto optimal front is quite common in general reinforcement learning problems [22].

The solution obtained from the pseudo brute force search is then used to linearly scale/normalize the value of each KPI, allowing the effective KPIs to be unitless and combined in a multi-objective optimization problem. The real goal of the system is to maximize the area spectral efficiency, energy efficiency, and user service rate while keeping the gap between target and achieved throughput values minimum. To be reflective of the real goals of the system, Eq. (7) is designed such that the normalized values of area spectral efficiency (between 0 and 1), energy efficiency (between 0 and 1), and user service rate (between 0 and 1) are multiplied in the numerator to jointly maximize these KPIs while the normalized value of throughput gap (between 1 and 2) is included in the denominator to minimize the difference between throughput obtained and achieved by the users. This gap-based formulation to model user satisfaction, instead of simple threshold based KPI where throughput is maximized for some users without a cap, is used as a clever way to avoid wasteful resource allocation. Compared to alternative simpler formulation where all KPIs are maximized as linear sum or product, this formulation is chosen to minimize intrinsic conflict QoS KPIs has with other two KPIs of area spectral efficiency and energy efficiency.

These four KPIs are representative of one of the four key aspects of network performance, either at the network level or user level. For instance, area spectral efficiency is representative of network spectral efficiency, energy efficiency is representative of network energy efficiency, user service rate is representative of scheduling maximum users while satisfying a certain data rate requirement, and throughput satisfaction is representative of meeting specific user throughput requirements. Note that using these many KPIs is not common in academia due to the intractability of the analytical models with complex multi-objective optimization functions. However, optimizing tens of KPIs simultaneously is a standard practice in real-time network optimization.

With the formulated optimization problem, a BBU controls the S-zone size of QoS categories such that desired KPIs (area spectral efficiency, energy efficiency, user service rate, and throughput satisfaction) are optimized while keeping the S-zone size within a specified range of R_{min} and R_{max}. The problem in Eq. (7) is a mixed-integer nonlinear programming problem with complexity of the order of $O\big((R_{max} - R_{min} + 1)^{|C|}\big)$. It is computationally difficult to achieve an optimal solution for a non-convex multi-objective problem in a dynamically changing network, which makes its application in real-time optimization systems impossible.

The legacy approaches to address such problem relies either on analytical modeling, or simulation-based modeling or more recently data-driven modeling. Our choice to leverage deep reinforcement learning instead of aforementioned approaches is motivated by its superiority to all three alternatives for the particular problem in hand. This superiority stems from the following reasons. Deep reinforcement learning-based framework is better than analytical model-based framework due to its ability to capture network dynamicity and complexity that analytical models miss to achieve due to the abstraction needed to obtain tractility. Compared to a simulator model-based offline optimization approach, deep reinforcement learning can tune optimization parameters of interest using live responses that reflect real-network behavior instead of an offline simulator behavior. Finally, deep reinforcement learning is advantageous compared to pure data-driven model-based optimization (e.g., using deep learning) as deep reinforcement learning does not require deluge of data that would be required to train a complex system-level network behavior data-driven model for performing the optimization. To this end, this paper proposes a D-RAN (DRL-based) framework that is capable of determining the optimal S-zone size for all QoS categories with the objective of maximizing network KPIs.

IV. Preliminaries

The following section gives a primer on deep reinforcement learning and simulated annealing algorithms.

A. Deep Reinforcement Learning

In a general reinforcement learning (RL) problem, an agent takes an action by observing the state from the environment and receives a scalar reward in an iterative manner. An RL agent aims to maximize the future cumulative rewards for different states of environments to learn the best course of action. Based on the specified set of actions, the RL algorithm generates a mapping between these actions and environment states. An implementation of RL includes these elements:

- *Observations*: Observations $\mathbf{O} \in \mathbb{R}^p$ are a set of measurements provided by the environment where p indicates the number of measurements observed.
- *States*: States $\mathbf{s}^t \in \mathcal{S}$ are a subset of observations vector observed at each epoch t either through handcrafted or non-handcrafted features where an epoch is a discretized time interval, signifying a single forward or backward pass of training samples.
- *Actions*: Actions $\mathbf{a}^t \in \mathcal{A}$ are a discrete/finite set of allowed choices that an RL agent can send to the environment as an input at each epoch t. Ideally, the choice of action should have an influence on the state of the environment such that the input of action changes the

state of the environment from $\mathbf{s}^t$ to $\mathbf{s}^{t+1}$.

- *Policy*: A policy $\pi(\mathbf{s}, \mathbf{a})$ is the mapping between the state of the environment and an agent's action.
- *Value function*: The value function (also called Q-function) under a given policy is given as $Q_\pi(\mathbf{s}, \mathbf{a})$ which represents the discounted future expected return for a state-action pair. The value function determines the value of being at a particular state and taking a specific action at that state [23].
- *Rewards*: The reward signal $r^{t+1} \in \mathbb{R}$ is a scalar value returned by the environment when an action $\mathbf{a}^t$ influences the state of the environment from $\mathbf{s}^t$ to $\mathbf{s}^{t+1}$.

These elements in conjunction drive the RL agent to maximize the future cumulative reward which is given as:

$$G = \sum_{t=0}^{\infty} \gamma^t r^{t+1}, \tag{8}$$

where $\gamma \in [0, 1]$ is the discount factor. Through iterative updates, the Q-function values are estimated using the Bellman equation in a traditional Q-learning algorithm:

$$Q^{t+1}(\mathbf{s}^t, \mathbf{a}^t) = (1 - \kappa) Q^t(\mathbf{s}^t, \mathbf{a}^t) + \kappa(r^{t+1} + \gamma \max_{\mathbf{a}} Q^t(\mathbf{s}^{t+1}, \mathbf{a}^{t+1})), \tag{9}$$

where $\kappa \in (0, 1]$ is the learning rate.

It is theoretically proven that Q-Learning algorithms converge under certain conditions [23]. However, the drawback of Q-learning is that it requires the agent to store a matrix of the size of state space times the size of action space, which is impossible for most real-world problems. To assuage that, deep neural networks (deep neural network) are utilized in RL algorithms, to act as universal Q-function approximators and learn the better representation of handcrafted features. The input dimension of deep reinforcement learning represents the number of states in the state space $|\mathcal{S}|$, while the output dimension represents the number of possible actions $|\mathcal{A}|$. The loss function with θ_Q as the trainable weights is used to train deep reinforcement learning is given below [24]:

$$\mathcal{L}(\theta_Q) = \mathbb{E}\big[r^{t+1} + \gamma \max_{\mathbf{a}^{t+1}} Q^t(\mathbf{s}^{t+1}, \mathbf{a}^{t+1}|\theta_Q) - Q^t(\mathbf{s}^t, \mathbf{a}^t|\theta_Q)\big]^2. \tag{10}$$

B. Simulated Annealing

The simulated annealing technique approximates the global optimum of nonlinear and non-convex objective functions by a series of iterative searches. Simulated annealing methodology is cognate to metallurgical annealing in which a metal is heated to a specific temperature before slowly cooling it down. simulated annealing begins its global optimum search with a very high-temperature parameter $Temp$, which enables it to explore a relatively wide area and then decreases the temperature, progressively narrowing the exploration area as it iteratively follows the steepest descent.

A fitness function associates a fitness value to each solution depending on the objective function. In each iteration, simulated annealing compares the fitness value of the current solution to the solutions that are available in the local neighborhood W. If the neighboring solution has a higher fitness value than the current solution, then the neighboring solution is chosen for the next iteration. The simulated annealing uses an acceptance probability to avoid adhering to a local optimum. The acceptance probability is given as follows [25]:

$$\text{Acceptance Probability} = \exp\left(-\frac{F_{curr} - F_{neig}}{Temp}\right), \forall\ neig \in W, \tag{11}$$

where F_{curr} represents the fitness value of current solution.

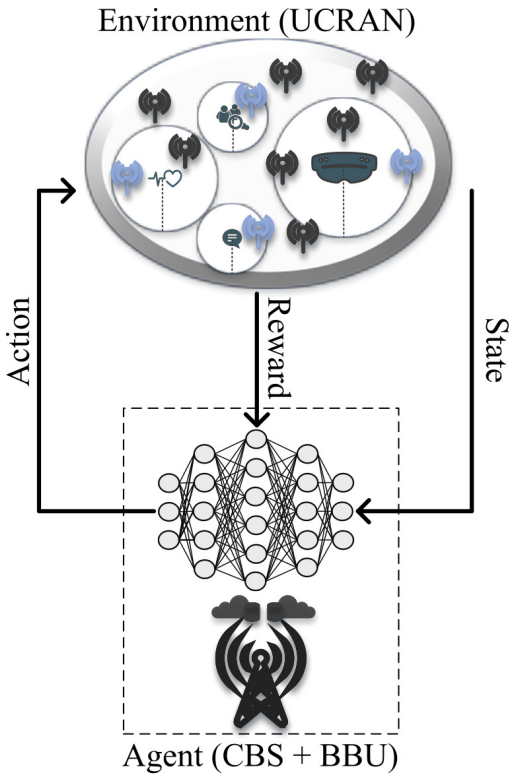


Fig. 3: Block diagram of the proposed D-RAN framework.

V. Proposed Solution

This section discusses the design of the proposed D-RAN framework. The multi-objective problem formulated in Eq. 7, even though a mixed-integer nonlinear programming problem with high complexity, can be solved using various optimization techniques including DRL-based approaches and meta-heuristics such as simulated annealing. To compare the effectiveness of proposed D-RAN framework (DRL-based approach) to a meta-heuristic approach, we have included a simulated annealing solution. As simulated annealing is also known to yield near optimal solutions for optimization problem of kind under consideration [26], it offers a benchmark to evaluate the performance of the proposed D-RAN framework in comparison to a state-of-the-art optimization solution approach. A BBU implements the optimization agent, which collects the network parameters and specifies the S-zone size for each QoS category. This centralized implementation facilitates the independence of processing times from UE and DBS densities, thus allowing for practical realizability and scalability of the optimization framework.

Algorithm 2: D-RAN Framework

Data: $\mathcal{A}, P, T, \eta, \epsilon, \epsilon_{max}, \epsilon_{min}, \epsilon_{decay}, E, R_{init}$
Initialize state, action, reward, and experience replay buffer $\mathcal{D}$;
while *converged or aborted* **do**
  $violate := 0$;
  Initialize S-zone size of QoS categories as $R_c := R_{init}\ \forall c \in C$;
  while $t \leq T$ **do**
    Observe environment state $\mathbf{s}^t$;
    $\epsilon := \max(\epsilon_{min}, \epsilon - (\epsilon_{max} - \epsilon_{min})\epsilon_{decay})$;
    if $z^t \sim U(0,1) < \epsilon$ **then**
      Select an action $\mathbf{a}^t \in \mathcal{A}$ randomly;
    else
      Select an action $\mathbf{a}^t = \arg\max_{\mathbf{a}^t} Q^t(\mathbf{s}^t, \mathbf{a}^t|\theta_Q)$;
    if $\mathbf{a}^t$ *violate* R_{min} *and* R_{max} *for any* R_c **then**
      Assign penalty P;
      $violate := violate + 1$;
      if $violate > \eta T$ **then**
        Abort the episode;
    Compute reward using Eq. (15);
    Observe next environment state $\mathbf{s}^{t+1}$;
    Store experience tuple $\{\mathbf{s}^t, \mathbf{a}^t, r^t, \mathbf{s}^{t+1}\}$ in the experience pool;
    Prioritize experiences using Eq. (16;
    Sample experiences in minibatch from $\mathcal{D}$ $e_y \triangleq \{\mathbf{s}^y, \mathbf{a}^y, r^y, \mathbf{s}^{y+1}\}$;
    Perform stochastic gradient descent on $\mathcal{L}(\theta_Q)$ given in Eq. (10);
    Update weight parameter θ_Q;
    $\mathbf{s}^t := \mathbf{s}^{t+1}$;

A. D-RAN Framework

A D-RAN framework is described in detail in terms of state space, action space, reward function, and the procedure of agent training and testing.

1) State Space

Section III-A establishes the linkage between the S-zone size of QoS categories and KPIs considered in this work. These KPIs define the state of the environment which if probed further can be decomposed into three parts:

- The average SINR of each QoS category is impacted by the change in S-zone size of QoS categories as divulged in Eq. (1), which has an impact on the area spectral efficiency, energy efficiency, user service rate, and throughput satisfaction. Increasing the S-zone size is expected to increase the average SINR inherently for two reasons: (i) a large S-zone yields a large minimal separation gap and hence reduction in interference between a scheduled UE and nearest interfering DBS; and (ii) a larger S-zone should lead to a higher macro-diversity gain due to selection among the larger number of DBSs in the S-zone. However, average SINR's impact on the listed KPIs makes it a suitable choice for defining environment state. The average SINR of each QoS category can be given as:

$$\varphi_c = \frac{\sum_{x \in N_c} \Gamma_x}{|N_c|}, \forall c \in C. \tag{12}$$

- The user service rate of each QoS category given in Eq. (5) determines the ratio of UEs from each QoS category that gets served, thus directly impacting the learning objective.
- The throughput satisfaction of each QoS category given in Eq. (6) relates to how well the achieved throughput compares to the throughput demanded by UEs in each QoS category. The high value of throughput satisfaction indicates a network overshooting or undershooting throughput, which requires some adjustment of S-zones.

In conjunction, the state vector of the proposed D-RAN framework with the cardinality of $3|C|$ is defined as:

$$\mathbf{s}^t = \{\varphi_1^t, ..., \varphi_{|C|}^t, \mathsf{U}_1^t, ..., \mathsf{U}_{|C|}^t, \mathsf{T}_1^t, ..., \mathsf{T}_{|C|}^t\}. \tag{13}$$

2) Action Space

For each QoS category, the action is to either increase or decrease the S-zone radius by d unit (measured in meters) or to keep it the same, that is, $\mathbf{a}_c = \{-d, 0, d\}$. Having a centralized agent responsible for adjusting the S-zone size for all QoS categories in the network will result in a combined action set.

The incremental action space has been selected to circumvent the combinatorically large action space that can be obtained by considering each combination of the QoS categories as an individual action, affecting the learning and convergence of the deep reinforcement learning agent greatly. Even with the incremental action space, the size of combined action space is $3^{|C|}$ for all QoS categories, which grows exponentially with QoS categories.

Motivated by the method to reduce deep reinforcement learning's large action space in [27], [28], the action space of each QoS category in D-RAN is considered as a separate action branch that controls an individual degree of freedom for each QoS category. By allowing individual action dimensions to operate independently, this approach ensures a linear increase in the size of combined action space with the number of QoS categories, of the order of $2|C| + 1$. For example, if $|C| = 2$, the following binary coding with $|C| + 1$ bits is used to represent the action space:

$$\mathbf{a} = \begin{cases} 101; & \text{increase } R_1 \text{ by } d \text{ meters.} \\ 001; & \text{decrease } R_1 \text{ by } d \text{ meters.} \\ 110; & \text{increase } R_2 \text{ by } d \text{ meters.} \\ 010; & \text{decrease } R_2 \text{ by } d \text{ meters.} \\ 000; & \text{keep } R_1 \ \& \ R_2 \text{ unchanged.} \end{cases} \tag{14}$$

In a similar way, the combined action space dimensionality reduction approach is scalable to networks with a greater number of QoS categories.

3) Reward Function

The reward function in D-RAN primarily focuses on two aspects for the S-zone size estimation in a dynamic environment: 1) finding the optimal trade-off between system-wide KPIs formulated as a multi-objective function given in Eq. (7),

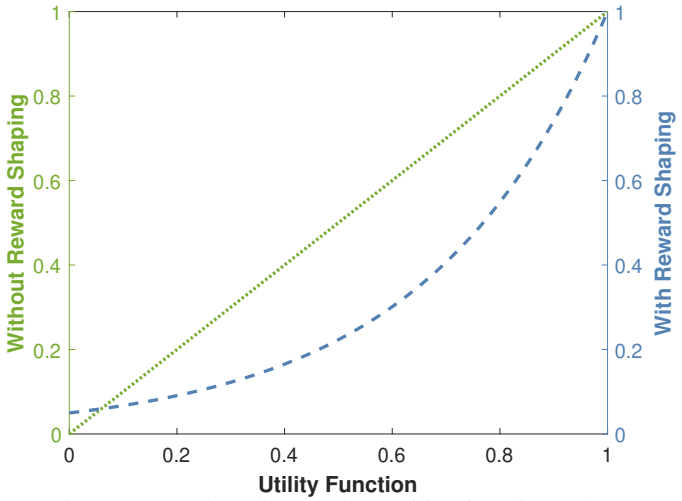


Fig. 4: Reward scaling against utility function values.

Algorithm 3: Simulated Annealing Framework

Data: $\mathcal{K}, N, R_{init}$
Initialize S-zone size of QoS categories as $R_c := R_{init}\ \forall c \in C$;
while $t \leq T$ **do**
  $current := \{R_1, R_2, ..., R_{|C|}\}$;
  Compute utility using Eq. (7) for $curr$;
  Append N neighboring solutions of $curr$ to **neigh** by choosing the adjacent combinations in $\mathcal{K}$;
  Compute utility using Eq. (7) for **neigh**;
  Compute acceptance probability AP using Eq. (11);
  if *acceptance probability of* i^{th} **neigh** $>$ $curr$ **then**
    $curr := \textbf{neigh}(i)$;
  else
    $curr := curr$

and 2) penalizing the agent for failure to satisfy the S-zone radius constraint given in Eq. (7). The utility function (u^t) at each TTI t is given as the objective function given in Eq. (7). Subsequently, the reward is calculated as follows:

$$r^t = \begin{cases} e^{\zeta(u^t-1)} & \text{if constraint given in Eq. (7) is met.} \\ Z & \text{otherwise,} \end{cases} \tag{15}$$

where $\zeta > 1$ in the exponential term is used to amplify the difference between values of the utility function and $-1 < Z < 0$ is a negative constant to punish the agent for choosing an S-zone size that is not within the specified bounds of R_{min} and R_{max}. The exponential shaping of the reward against utility values allows the deep reinforcement learning agent to give a much higher reward when it achieves higher utility values and much lesser when it achieves lesser or mid-range utility values, as shown in Fig. 4. In contrast, the linear shaping of reward has a higher reward even for mid-range utility values, which may tempt the reinforcement learning agent to choose sub-optimal actions. The reward function is designed to obtain values between -1 and 1 to accelerate the stochastic gradient descent algorithm in the deep neural network [29], [30].

4) Agent Training & Testing Procedure

The schematic diagram of the proposed D-RAN framework is shown in Fig. 3. The learning agent located in BBU collects state information from the environment and aims to find the optimal action policy (S-zone size for all QoS categories) such that the reward function given in Eq. (15) is maximized. The deep neural network includes four fully connected layers, and three rectified linear unit activation functions with input layer neurons equal to the number of state variables $3|C|$ and output layer equivalent to the number of actions $2|C| + 1$.

As part of the training process, the agent stores the experience tuple $\{\mathbf{s}^t, \mathbf{a}^t, r^t, \mathbf{s}^{t+1}\}$ in the experience pool with buffer size $\mathcal{D}$ and updates the deep neural network weights in Eq. (10) by applying the stochastic gradient descent algorithm to a minibatch of data at each epoch t (equivalent to a TTI) as detailed in Algorithm 2. As part of the execution/testing process, the agent collects the state information from the environment and outputs the action in each TTI. In every episode, consisting of T epochs/TTIs, the agent is initialized at R_{init} for all QoS categories, and the environment is initialized with different random seeds to generate different mobility patterns. An episode is ended prematurely only if the agent chooses S-zone of any QoS category that is beyond the allowed limits of S-zone size (R_{min} and R_{max}) for more than ηT times, where $0 \leq \eta \leq 1$ is a design parameter used to limit the proportion of wrong actions to ensure that the agent learns "what not to learn" [31].

The experiences drawn from experience replay during training are prioritized according to the importance of the tuple, which is dependent on the temporal difference that measures the unexpected deviation from the state transition value [32]. The prioritized experience replay algorithm stores the subsequent temporal difference error with each state transition and assigns high priority to experiences that have high temporal difference error and are recent. A stochastic sampling method is used in the D-RAN framework to interpolate experience samples between greedy and uniform random sampling by using the following formula:

$$Y = \frac{p_y^{\upsilon}}{\sum_z p_z^{\upsilon}}, \tag{16}$$

where $p_y > 0$ is the priority of transition y and the exponent υ determines the prioritization weightage, with $\upsilon = 0$ corresponding to the uniform random sampling. The prioritized experience replay model ensures stability and avoids local minimum convergence. To further assist stability in D-RAN training, a target deep neural network is used to predict the target Q-values that are updated after every U steps.

D-RAN adopts an exploration algorithm with the exploration variable ϵ initialized at ϵ_{max} and decayed linearly at a rate of ϵ_{decay} until ϵ_{min} is reached. If the current exploration rate ϵ is greater than a random uniform distribution sample, then the deep reinforcement learning agent chooses a random action. Learning is deemed to have converged when the average reward function is flat and no longer increases in the last E episodes. The Algorithm 2 steps can be summarized as follows:

- Initialize the environment and agent parameters.

TABLE II: Network simulation and training parameters.

Symbol	Parameter Name	Parameter Value
λ_{UE}	UE average density	$10^3 \backslash km^2$
λ_{DBS}	DBS average density	$10^3 \backslash km^2$
PLE	Path-loss exponent	3
R_{min}	Minimum S-zone size	$10\,m$
R_{max}	Maximum S-zone size	$80\,m$
R_{init}	Initial S-zone for each QoS category	$(R_{max} + R_{min})/2\,m$
d	Action space stepsize	$3\,m$
α, β	Weightage parameters in Eq. (7)	$0.4, 0.4$
P	Penalty for wrong action	-1
T	Number of epochs/TTIs	1000
η	Percentage of wrong actions allowed	5
ϵ_{max}	Maximum exploration rate	1.0
ϵ_{min}	Minimum exploration rate	0.1
ϵ_{decay}	Exploration rate decay	$0.0002/\|C\|$
U	Target deep neural network update epochs	50
E	Convergence episodes	50
N	Number of neighbor solutions	2

- Observe the state of the environment at TTI t.
- Select the action at TTI t.
- Compute the reward for the action taken based on Eq. (15).
- Train the prioritized experience replay with the experience tuples.
- Repeat the above steps until learning has converged or aborted.

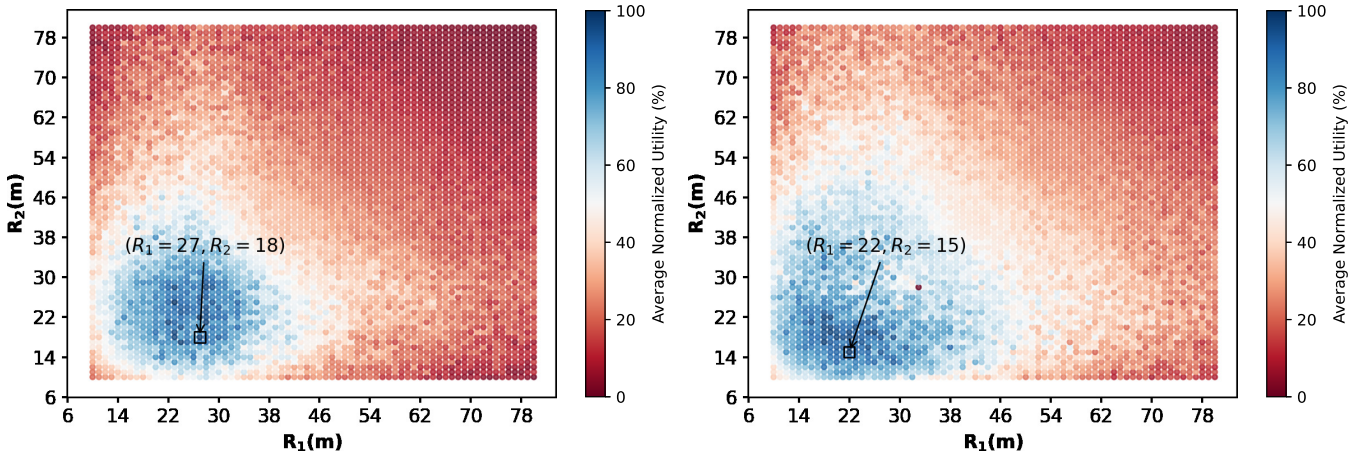
B. Simulated Annealing Framework

Implementing a meta-heuristic such as simulated annealing for S-zone optimization in principle is similar to implementing a D-RAN framework, as the optimization agent is embedded in the BBU that adjusts the size of S-zones for all QoS categories. Instead of observing the environment state, the simulated annealing algorithm takes into account the current solution, defined as the concatenation of S-zones sizes of all QoS categories; thus, $curr = \{R_1, R_2, ..., R_{|C|}\}$. The simulated annealing algorithm traverses several neighboring solutions at each TTI and calculates the fitness of each of them. The neighboring solution space is derived from the entire solution search space $\mathcal{K}$ that includes the combinations of allowed S-zone size of all QoS categories such that its size will be $(\frac{R_{max}-R_{min}+1}{d})^{|C|}$. As such, the neighboring search space will be defined as the S-zones combinations that are adjacent to the current solution in $\mathcal{K}$. If the utility value of the neighboring solution is greater than the current solution or its acceptance probability is greater than a certain threshold, the neighboring solution is accepted. The acceptance probability is calculated using the formula given in Eq. (11) which is sensitive to temperature parameter $Temp$ with the fitness function is equivalent to the utility function given in Eq. (7) as detailed in Algorithm 3.

VI. Experimental Evaluation

Unlike the physical layer, not much data can be gathered to build pure data-driven models for system-level optimization problems. This is mainly because: 1) network operators cannot afford to try all the parameter ranges in a live network for empirical data generation, and 2) real-network data is not currently available because novel architectures, such as the user-centric architecture investigated in this paper, are still a concept that will be implemented in 6G and beyond networks. While D-RAN does not require deluge of data from live network before-hand for training an explicit and static network behavior model, it does require some interaction on the live network, or some data from the network to build an implicit dynamic sketch of the model. As, no UCRAN-based 6G or beyond network yet exist, we resort to a system-level simulator to meet this requirement. Although we use simulator-generated data in this study to train D-RAN, the insights gained remain valid for real scenarios, when the proposed D-RAN will be eventually built using data from a live network, once the proposed architecture shows benefit and is deployed in real-networks. Even in that case, pre-training the D-RAN using synthetic data from a simulator and then fine-tuning the model from live network data might be needed to address the data scarcity challenge, making the proposed synthetic data-aided deep reinforcement learning training approach worthy of investigation.

This section presents the performance of proposed D-RAN framework with system model presented in Section II. The target coverage area of CBS is 1 square kilometer. The UEs and DBSs are distributed through an homogeneous Poisson point processes within the CBS coverage region. This work considers a maximum of three QoS categories with throughput and latency requirements of 1: virtual/augmented reality, 2: E-health, and 3: monitoring sensor networks, respectively. The number of QoS categories is determined by the network operator depending on the dominant traffic types in a specific CBS coverage area. The minimum and maximum S-zone size considered in this work are 10 meters and 80 meters, respectively with the action space step size of 3 meters. The



(a) (b)

Fig. 5: Comparison of brute-force solution for different UE placement realizations in a two-dimensional S-zone space.

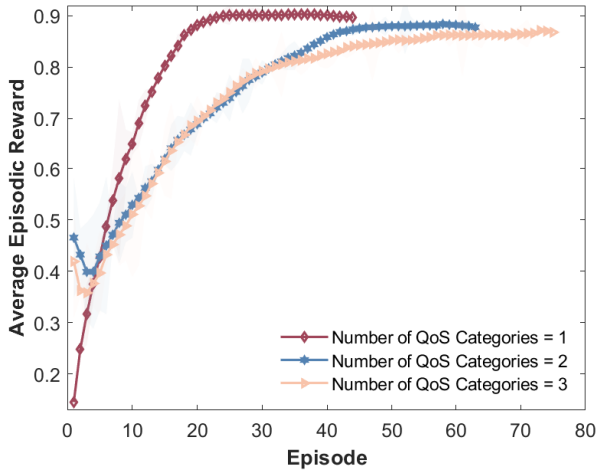


Fig. 6: Convergence of the average episodic reward values for varying number of QoS categories. To improve readability, these curves are smoothed with a moving average taken over 20 episodes. The shade represents the standard deviation.

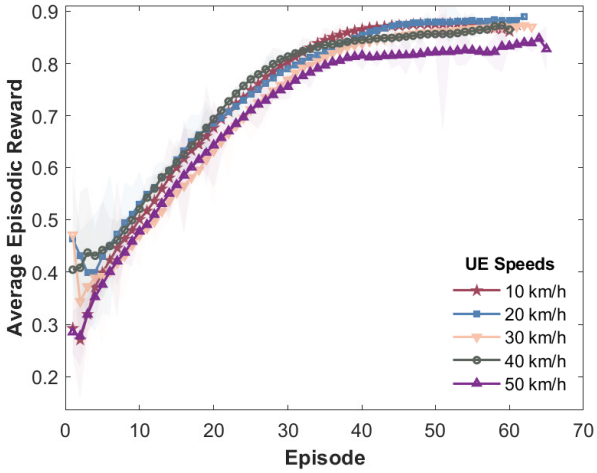


Fig. 7: Convergence of the average episodic reward values for varying maximum UE speeds. To improve readability, these curves are smoothed with a moving average taken over 20 episodes. The shade represents the standard deviation.

choice of these user-centric cells (S-zone) size limits are inspired by industry standards [33].

Python 3.6 and Pytorch are utilized to conduct these experiments. The number of maximum epochs / TTIs (T) in each training and evaluation episode is set to 1000, where each TTI's duration is set to 1 ms. Both deep neural networks used in the main and the target network have three hidden layers containing 128-256-128 neurons. A careful choice of depth and width of these deep neural networks is made to avoid underfitting or overfitting of the nonlinear mapping between inputs and outputs. The size of the minibatch for deep neural network training is set to 64, and the target network is updated after every 50 TTIs. The rest of the network parameters and hyperparameters required to tune deep reinforcement learning-assisted and simulated annealing-assisted frameworks are shown in Table II.

A. Brute-Force Solution

The brute-force S-zone selection attempts to solves the optimization problem given in Eq. (7) by exhaustively searching the S-zone space of size $(\frac{R_{max}-R_{min}+1}{d})^{|C|}$. With the considered values of R_{max} and R_{min}, the brute-force solution may be a feasible option if the size of search space is less than a million combinations ($|C| < 4$). However, the size of S-zone space is not the only deterrent in making a brute-force solution infeasible. UE mobility have a direct effect on SINR, which in turn impacts the KPI values used in the utility function in Eq. (7), making a static solution for S-zone selection infeasible due to its complexity of the order of $O\left((\frac{R_{max}-R_{min}+1}{d})^{|C|\times T}\right)$.

Fig. 5a and Fig. 5b shows the averaged normalized utility function for the different realizations of UEs positions for $|C| = 2$. While the concave envelope of maximum utility is somewhat maintained in the Fig. 5 (blue region), the individual utility values corresponding to each S-zone size

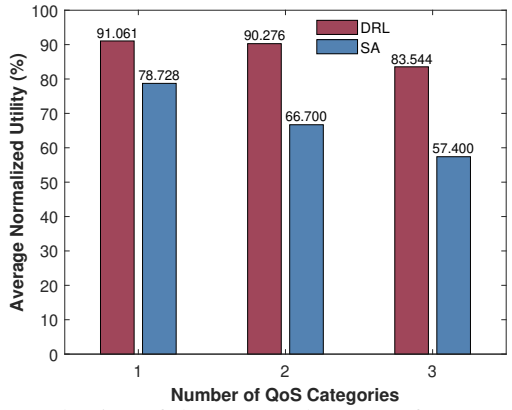


Fig. 8: Evaluation of the proposed D-RAN framework against simulated annealing framework for maximum UE speeds equal to 10 km/h.

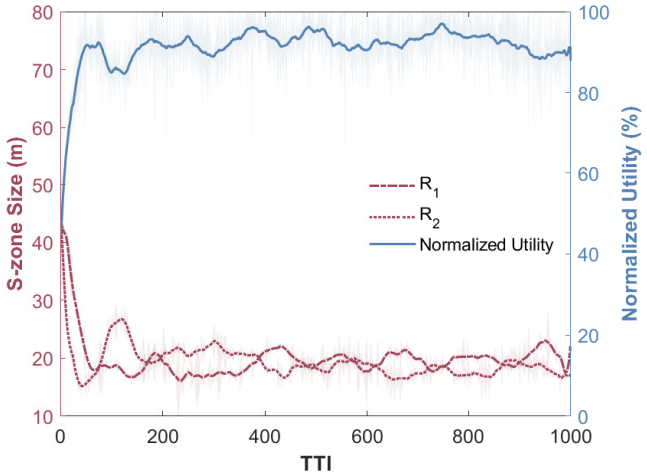


Fig. 9: Proactive real-time S-zone size optimization for $|C| = 2$. To improve readability, these curves are smoothed with a moving average taken over 50 TTIs. The shade represents the standard deviation.

combination as well as the apex of the utility function is shown to change. For example, the maxima of utility function (black square) changes from $(R_1 = 27, R_2 = 18)$ in Fig. 5a to $(R_1 = 22, R_2 = 17)$ in Fig. 5b. Because UE mobility follows random way point model, these values may change in each TTI, which makes it necessary to assign S-zone sizes to the QoS categories dynamically and intelligently by interacting with the environment. To this end, deep reinforcement learning is a more appropriate choices in solving non-deterministic and real-time optimization of S-zone sizes.

B. Convergence Comparison for Varying Number of QoS Categories

The convergence of the proposed D-RAN framework with dynamicity in the network due to heterogeneous user application demands is shown for different numbers of QoS categories in Fig. 6. The value of the utility function is normalized with the upper and lower limits, determined by the brute-force solution so that the reward function can have a maximum and minimum value of 1 and -1, respectively. For each of the considered cases in Fig. 6, the learning converges towards higher reward function values after a certain amount of training episodes. The greater the number of QoS categories, the longer it takes to converge due to a larger state space, action space, and search space, requiring more TTIs to explore the environment. Additionally, as the number of QoS categories increases, the reward function tends to converge to a lower reward value. This is mainly due to the expansion of S-zone space and the increase in the minimum required number of TTIs to reach to optimal S-zone (R_c^*) from the initial S-zone (R_c^{init}) for each QoS category.

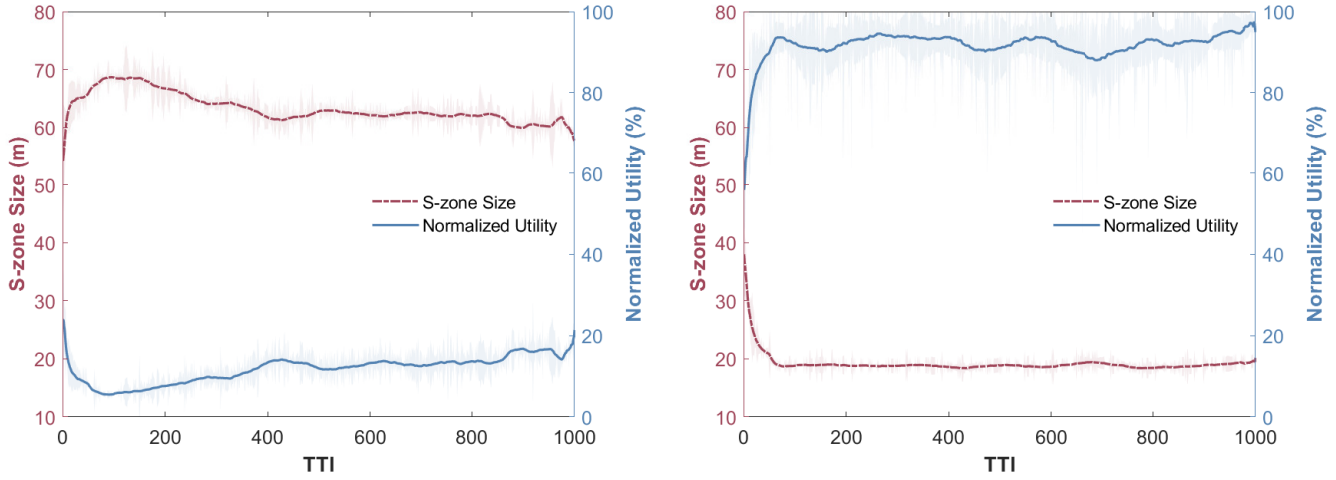
C. Convergence Comparison for Non-stationary UEs

Fig. 7 shows the convergence of D-RAN framework with varying maximum UE mobility speeds for $|C| = 2$. In each episode, a different random seed is used in the random waypoint mobility model, effectively changing the mobility pattern of UEs allowing the agent to learn the dynamics of the environment. The purpose of training with non-stationary UEs distribution is to determine whether the D-RAN framework can dynamically adjust S-zone size as the distribution of UEs changes. In the figure, it can be seen that the reward function tends to converge for each of the considered cases with a decrease in steadiness as the UE speed increases. This is mainly because the higher the UE speed, the more significant the change in the user distribution, leading to highly non-stationary maxima of the utility function causing the oscillations in convergence. However, the reward function on average converges to higher reward values, with the gap between converged reward values and maximum possible reward depicting the minimum required number of TTIs to reach to optimal S-zone (R_c^*) from the initial S-zone (R_c^{init}) as discussed in Section VI-B.

D. Evaluation for Different QoS Categories

In this experiment, the proposed D-RAN and simulated annealing-assisted frameworks are tested for varying number of QoS categories. The D-RAN framework is evaluated using the trained weights (state-action mapping), while the simulated annealing-assisted framework is evaluated using heuristic optimization. Performance is measured by averaging 1000 TTIs for 100 testing scenarios based on the utility function given in Eq. (7). To compare the performance in relative terms to maximum achievable utility, the utility values are normalized from maximum and minimum utility values obtained from the brute-force solution.

Compared to a brute-force solution that requires large computations and cannot scale, the D-RAN framework exhibits better adaptability to changing environmental conditions and maintains utility at a near-optimal level, as shown in Fig. 8. Additionally, the D-RAN framework surpasses the performance of the simulated annealing-assisted framework due to the slow convergence of simulated annealing optimization and the high sample complexity required to reach a reasonable solution if the search space is too large. Fig. 8 illustrates this phenomenon, where simulated annealing performances decrease as the number of QoS categories and the combinatorial search space increase. On the other hand, D-RAN



(a) Exploration stage.

(b) Exploitation stage.

Fig. 10: TTI-wise normalized utility comparison for exploration and exploitation stages of D-RAN training for $|C| = 1$. To improve readability, these curves are smoothed with a moving average taken over 50 TTIs. The shade represents the standard deviation.

framework manage to maintain a level of uniformity in terms of average utility scores across the QoS categories due to their ability to solve combinatorial optimization problems. Note that the D-RAN framework is not learning on the channel fading values directly since predicting/learning channel fading is too complex a task for any learning framework, particularly at a short time scale at which fast fading changes. However, the channel fading does introduce randomness in the state values and reward function of the D-RAN agent, which it considers as a random perturbation of the environment. The deep reinforcement learning agents have generally been shown to better explore the environment with random perturbations caused due to the slight imperfection of the state values or reward function. The authors in [34] have also made a similar observation where the deep reinforcement learning agent observing noisy reward sometimes even outperforms the case with the true reward, which they attribute to the implicit exploration introduced by the perturbations in the reward.

E. Proactive Real-time S-zone Optimization

The epoch / TTI-wise S-zone size optimization is shown in Fig. 9. To maximize the utility function, the proposed D-RAN framework adjusts the S-zone size for each QoS category to obtain the Pareto-optimal solution for area spectral efficiency, energy efficiency, user service rate, and throughput satisfaction. The S-zone size for each QoS category begins with an initial S-zone size of $\frac{R_{max}-R_{min}}{2} = 45\,m$ and then move towards the near-optimal S-zone size for each category as shown in Fig. 9. It can be observed that D-RAN continuously adjusts S-zone size of each QoS category with the changing network dynamics resulting in the maximization of the normalized utility. Fig. 10 shows the changes in S-zone size for $|C| = 1$ with associated utility scores during the exploration and exploitation stages of D-RAN. In the exploration stage, the D-RAN agent explores the environment by executing random actions so as to gain knowledge of it as shown in Fig. 10a. While in the exploitation stage, the agent uses its current knowledge (deep neural network weights and state-action mapping) to change S-zones size to gain higher rewards as shown in Fig. 10b. The results in Fig. 10b show that the utility function is higher (near-optimal) in the exploitation stage, indicating good learning of state-action mapping of the environment.

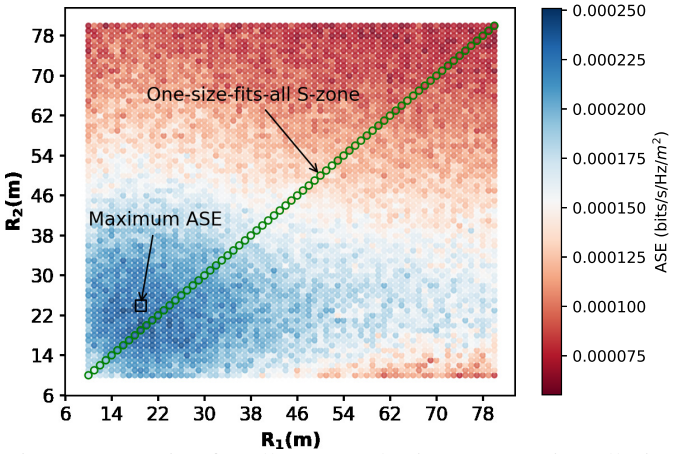
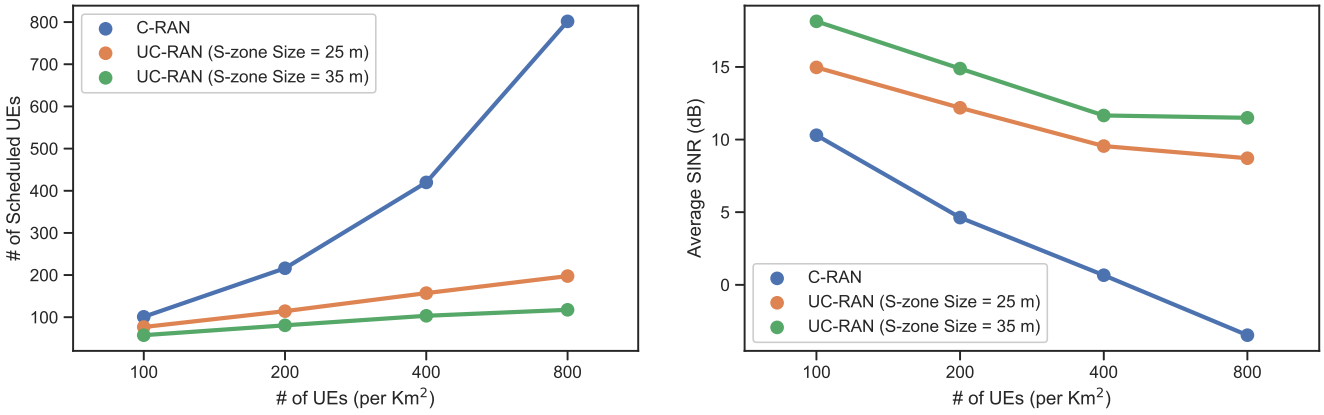


Fig. 11: One-size-fits-all versus elastic user-centric cell size comparison for area spectral efficiency.

F. S-zone's Elasticity Impact on area spectral efficiency

Fig. 11 compares the one-size-fits-all S-zone size (green circles) and elastic S-zone size for $|C| = 2$. This result supports the claim in Section I-C that assigning the same S-zone to all categories may not be optimal for accommodating heterogeneous throughput and latency requirements. The figure shows that the maximum achievable area spectral efficiency (black square) does not even lie within a one-size-fits-all S-zone space. The elastic S-zone architecture, however, allows for adaption to heterogeneous QoS requirements, which



(a) Number of scheduled UEs. (b) Average SINR (dB).

Fig. 12: Comparison of user-centric (UC-RAN) and non-user-centric (C-RAN) networks.

maximizes area spectral efficiency for the whole network.

G. *Comparison of User-centric with Non-user-centric architecture*

To compare the performance of the proposed user-centric approach with a non-user-centric approach, we simulate a Cloud Radio Access Network (C-RAN) model which considers similar assumptions as taken for a user-centric architecture to ensure a fair comparison between the two architectures. These assumptions are: (i) the DBSs are deployed in high density, (ii) each UE is allocated the full bandwidth of the system, (iii) there is a one-to-one association between UE and DBS, and (iv) the UE is associated with a DBS providing the maximum channel gain. With these assumptions, the only contrasting factor in C-RAN and UC-RAN architectures is the S-zone parameter which ensures minimal separation between the scheduled UEs.

Fig. 12 shows the average SINR and number of scheduled UEs plots for varying UE densities. It can be observed that the average SINR in the case of C-RAN falls drastically with the increase in the density of UEs in the network. At the same time, UC-RAN architecture with the additional degree of freedom (S-zone size) is able to achieve much higher average SINRs at the cost of lesser scheduled UEs. The S-zone size controls the separation between the scheduled UEs, impacting the average SINR and the number of scheduled UEs. From Fig. 12, it can be hypothesized that the C-RAN (traditional Heterogenous network) architecture will not be able to perform better in a network with dense DBS deployment, which is envisaged for 6G and beyond networks. On the other hand, the UC-RAN architecture can provide an effective solution to this problem by incorporating an additional degree of freedom (S-zone size). Manually selecting the S-zone size will only be applicable if the environment is not dynamic and the solution space is too small. Therefore, intelligent control of S-zone size is needed to optimally choose the S-zone size in a dynamic environment with more than one QoS category.

VII. Conclusion

In this paper, we proposed D-RAN: a deep reinforcement learning-based user-centric RAN optimization framework under dynamic user application demands and network conditions. Unlike previous cellular network approaches, D-RAN employs a concept of elasticity within user-centric systems that employ non-uniform virtual cells (also called S-zones) for different QoS categories (e.g., Augmented/Virtual Reality and E-health applications). To avoid searching exhaustively using brute-force or meta-heuristics, a D-RAN framework has been developed to adjust S-zone sizes based on changing network dynamics such as user mobility. D-RAN introduces a less complex approach than brute-force or meta-heuristic techniques by accurately learning the mapping of environmental conditions to S-zone size of corresponding QoS categories. A multi-objective problem is optimized in real-time in the proposed architecture based on KPIs like area spectral efficiency, energy efficiency, UE service rate, and throughput satisfaction. Simulated results indicate that D-RAN framework is nearly as effective as brute-force and surpasses meta-heuristics like simulated annealing, but with lower complexity and is adaptable to dynamic changes in the network. In general, this research aims to introduce intelligence into user-centric elastic networks to accommodate user applications' non-uniform throughput and latency requirements. With the proposed D-RAN framework, the paradigm of traditional cellular networks could be transformed into demand-driven, elastic, user-centric systems in future 6G and beyond networks.

Acknowledgements

This work is supported by the National Science Foundation under Grant Numbers 1923669 and 1923295, and Qatar National Research Fund under Grant No. NPRP12-S 0311-190302. For more inquiries about these projects please visit www.AI4networks.com.

References

[1] B. Romanous, N. Bitar, A. Imran, and H. Refai, "Network densification: Challenges and opportunities in enabling 5g," in *2015 IEEE 20th*

International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD). IEEE, 2015, pp. 129–134.

[2] Y. Yang, K. W. Sung, J. Park, S.-L. Kim, and K. S. Kim, "Cooperative transmissions in ultra-dense networks under a bounded dual-slope path loss model," in *2017 European Conference on Networks and Communications (EuCNC)*. IEEE, 2017, pp. 1–6.

[3] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6G be?" *Nature Electronics*, vol. 3, no. 1, pp. 20–29, 2020.

[4] Z. Cheng, D. Zhu, Y. Zhao, and C. Sun, "Flexible Virtual Cell Design for Ultradense Networks: A Machine Learning Approach," *IEEE Access*, vol. 9, pp. 91 575–91 583, 2021.

[5] U. S. Hashmi, S. A. R. Zaidi, and A. Imran, "User-centric cloud RAN: An analytical framework for optimizing area spectral and energy efficiency," *IEEE Access*, vol. 6, pp. 19 859–19 875, 2018.

[6] Y. Zhang, B. Di, H. Zhang, J. Lin, C. Xu, D. Zhang, Y. Li, and L. Song, "Beyond cell-free mimo: Energy efficient reconfigurable intelligent surface aided cell-free mimo communications," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 2, pp. 412–426, 2021.

[7] U. S. Hashmi, S. A. R. Zaidi, A. Imran, and A. Abu-Dayya, "Enhancing downlink QoS and energy efficiency through a user-centric Stienen cell architecture for mmWave networks," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 387–403, 2020.

[8] K. Humadi, I. Trigui, W.-P. Zhu, and W. Ajib, "Dynamic Base Station Clustering in User-Centric mmWave Networks: Performance Analysis and Optimization," *IEEE Transactions on Communications*, 2021.

[9] S. K. Kasi, U. S. Hashmi, M. Nabeel, S. Ekin, and A. Imran, "Analysis of Area Spectral & Energy Efficiency in a CoMP-Enabled User-Centric Cloud RAN," *IEEE Transactions on Green Communications and Networking*, 2021.

[10] N. Guo, M.-L. Jin, and N. Deng, "Coverage analysis for heterogeneous network with user-centric cooperation," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2724–2727, 2018.

[11] S. K. Kasi, U. Sajid Hashmi, M. Nabeel, S. Ekin, and A. Imran, "Is comp beneficial in user-centered wireless networks?" in *2022 1st International Conference on 6G Networking (6GNet)*, 2022, pp. 1–5.

[12] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE network*, vol. 28, no. 6, pp. 27–33, 2014.

[13] A. Mohamed, O. Onireti, M. A. Imran, A. Imran, and R. Tafazolli, "Control-data separation architecture for cellular radio access networks: A survey and outlook," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 446–465, 2016.

[14] A. Taufique, M. Jaber, A. Imran, Z. Dawy, and E. Yacoub, "Planning wireless cellular networks of future: Outlook, challenges and opportunities," *IEEE Access*, vol. 5, pp. 4821–4845, 2017.

[15] S.-F. Cheng, L.-C. Wang, C.-H. Hwang, J.-Y. Chen, and L.-Y. Cheng, "On-device cognitive spectrum allocation for coexisting urllc and embb users in 5g systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 171–183, 2021.

[16] K. Khawam, S. Lahoud, M. E. Helou, S. Martin, and F. Gang, "Coordinated framework for spectrum allocation and user association in 5g hetnets with mmwave," *IEEE Transactions on Mobile Computing*, vol. 21, no. 4, pp. 1226–1243, 2022.

[17] U. S. Hashmi, A. Rudrapatna, Z. Zhao, M. Rozwadowski, J. Kang, R. Wuppalapati, and A. Imran, "Towards real-time user qoe assessment via machine learning on lte network data," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–7.

[18] L. Sboui, Z. Rezki, A. Sultan, and M.-S. Alouini, "A new relation between energy efficiency and spectral efficiency in wireless communications systems," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 168–174, 2019.

[19] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume *et al.*, "How much energy is needed to run a wireless network?" *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40–49, 2011.

[20] H. Khaled, I. Ahmad, D. Habibi, and Q. V. Phung, "A green traffic steering solution for next generation communication networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 222–238, 2021.

[21] A. R. Ramos, B. C. Silva, M. S. Lourenço, E. B. Teixeira, and F. J. Velez, "Mapping between average sinr and supported throughput in 5g new radio small cell networks," in *2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2019, pp. 1–6.

[22] P. Mannion, S. Devlin, J. Duggan, and E. Howley, "Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning," *The Knowledge Engineering Review*, vol. 33, 2018.

[23] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[24] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[25] S. K. Kasi, M. K. Kasi, K. Ali, M. Raza, H. Afzal, A. Lasebae, B. Naeem, S. ul Islam, and J. J. Rodrigues, "Heuristic edge server placement in Industrial Internet of Things and cellular networks," *IEEE Internet of Things Journal*, 2020.

[26] K. Amine, "Multiobjective simulated annealing: Principles and algorithm variants," *Advances in Operations Research*, vol. 2019, 2019.

[27] A. Tavakoli, F. Pardo, and P. Kormushev, "Action branching architectures for deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[28] F. Wei, G. Feng, Y. Sun, Y. Wang, S. Qin, and Y.-C. Liang, "Network slice reconfiguration by exploiting deep reinforcement learning with large action space," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2197–2211, 2020.

[29] X. Tao and A. S. Hafid, "Deepsensing: A novel mobile crowdsensing framework with double deep Q-network and prioritized experience replay," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 547–11 558, 2020.

[30] S. K. Kasi, S. Das, and S. Biswas, "TCP Congestion Control with Multi-agent Reinforcement and Transfer Learning," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2021, pp. 1507–1513.

[31] T. Zahavy, M. Haroush, N. Merlis, D. J. Mankowitz, and S. Mannor, "Learn what not to learn: Action elimination with deep reinforcement learning," *arXiv preprint arXiv:1809.02121*, 2018.

[32] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. Van Hasselt, and D. Silver, "Distributed prioritized experience replay," *arXiv preprint arXiv:1803.00933*, 2018.

[33] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, "5G evolution: A view on 5G cellular technology beyond 3GPP release 15," *IEEE access*, vol. 7, pp. 127 639–127 651, 2019.

[34] J. Wang, Y. Liu, and B. Li, "Reinforcement learning with perturbed rewards," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6202–6209.