Reinforcement Learning for Automated Energy Efficient Mobile Network Performance Tuning

Diarmuid Corcoran
Ericsson AB and Software and Computer Systems
KTH Royal Institute of Technology
Stockholm, Sweden
diarmuid.corcoran@ericsson.com

Per Kreuger
RISE AI
Research Institutes of Sweden
Kista, Sweden
per.kreuger@ri.se

Magnus Boman
Software and Computer Systems
KTH Royal Institute of Technology
Stockholm, Sweden
mab@kth.se

***Abstract*—Modern mobile networks are increasingly complex from a resource management perspective, with diverse combinations of software, infrastructure elements and services that need to be configured and tuned for correct and efficient operation. It is well accepted in the communications community that appropriately dimensioned, efficient and reliable configurations of systems like 5G or indeed its predecessor 4G is a massive technical challenge. One promising avenue is the application of machine learning methods to apply a data-driven and continuous learning approach to automated system performance tuning. We demonstrate the effectiveness of policy-gradient reinforcement learning as a way to learn and apply complex interleaving patterns of radio resource block usage in 4G and 5G, in order to automate the reduction of cell edge interference. We show that our method can increase overall spectral efficiency up to 25% and increase the overall system energy efficiency up to 50% in very challenging scenarios by learning how to do more with less system resources. We also introduce a flexible phased and continuous learning approach that can be used to train a bootstrap model in a simulated environment after which the model is transferred to a live system for continuous contextual learning.**

***Index Terms*—Communication system traffic, Machine learning, Learning systems, System simulation, Self-organization, Radio resource scheduling, Inter-cell interference coordination**

I. Introduction

With the evolution of mobile systems such as LTE [1] and introduction of 5G Radio Access Networks (RAN) [2] managing and organizing these systems has become a huge technical and financial challenge. There is thus a need for more intelligent self-organizing and self-optimizing system solutions. For LTE, the concept of SON (Self Organizing Networks) [3] was introduced with the intent to reduce manual effort and increase network automation. While a step in the right direction, it has been shown that the current SON approach is limited and that considerable challenges exist for SON in 5G using existing approaches [4].

One very promising development strand is data-driven [5] and machine learning approaches to aid with complex optimization problems [6], [7]. The synthesis of these techniques with simulation and RAN internal data distribution mechanisms provides the basis for creating a whole new generation of intelligent algorithms [8]. Reinforcement learning (RL) [9] has shown huge potential [10] in learning game control strategies but also in real-world applications such as antenna tilt optimization [11], routing [12], data-center energy management [13] and robotic control [14]. In RAN self-management and control, work is starting to emerge (see Section II) but is still limited, mostly offline [15], and with little experimental data available suggesting a need to explore the potential of RL approaches further (cf. [16]).

In this paper, we apply the concept of continuous RL to learn effective solutions to the SON problem of Inter-Cell Interference Coordination (ICIC) [17]. Both LTE and 5G support Orthogonal Frequency Division Multiple Access (OFDMA) [18] where all cells/sectors/beams can share the same set of frequency resources to allow higher spectral efficiency, i.e. a frequency reuse of one. This flexibility, however, leads to interference between allocated radio resources at cell boundaries, see Figure 1 for an overview. The problem of interfering cells in a RAN system becomes particularly difficult with the deployment of 5G and increasingly dense small cell deployments, heterogeneous networks (HetNet) with mixed high and low power cells, and even *ad hoc* configurations [19]. With this challenge in mind, detailed manual cell planning becomes difficult, time consuming and expensive, and the need for self-organizing approaches ever more important.

In our continuous learning approach, we use a simulation environment and an RL approach to train an interference reduction policy. The policy can then be transferred to a real RAN system where further, sample based, contextual learning can continue. The simulation environment allows for more than just local optimization by also considering feedback effects resulting from the RL agent's efforts: how the learned changes propagate through the network.

Key Contributions

- An RL method to derive scheduling and allocation policies to balance system throughput, energy utilization, and user fairness.
- Clear and significant gains in terms of measured performance and energy savings.
- Extensive empirical measurements demonstrating the method's potential in very challenging network scenarios.
- An incremental approach, using scheduling agents pretrained in a simulation environment, for later transfer to live systems and further on-line contextual training.

978-3-903176-36-2 ©2021 IFIP

Disposition: We first describe the ICIC problem and its state-of-the-art, before turning to our own model and the assumptions associated with it in Section III. In Section IV, we describe and discuss our experiments and empirical results, before Section V concludes the paper with a discussion on the applicability of this work and possible future extensions.

II. Problem Description and Related work

Figure 1-(a) depicts ICIC: users at cell edges are subjected to increased interference due to cells' reuse of radio resources. Figure 1-(b) shows a radio frame structure valid for both LTE and 5G[1]. Each 1 ms time slot is called a subframe and are separated in the frequency domain into chunks of 180Khz, and together, are called physical resource blocks (PRBs) (see Figure 1-(c)), across a total system bandwidth which depends on specific configuration and radio standard: LTE or 5G. For ultra-dense and *ad hoc* small cell configurations, the inter cell distance decreases and the overlap zones increase, leading to increased inter-cell interference and more devices falling into these overlap zones. To deal with these increasingly densified small cell and HetNet configurations, we need better cooperative and self-organizing mechanisms to determine detailed radio resource allocation strategies and we propose one such approach in this work.

In 4G and 5G RAN systems, there are currently four main approaches [17], [21] to solving ICIC: 1) In the frequency domain by using various static or dynamic frequency separation strategies; 2) In the time domain by coordinating use of transmitted radio resource at subframe level; 3) In the power domain, by adjusting the transmit power or using a range extension mechanism in one or more interfering base stations; 4) Tight coordinated multi-point (CoMP) transmission within a cluster of cells. Each of these approaches has been well studied, with many reports on various algorithms and optimization techniques, both centralized and distributed [19], [22], [23].

In previous work, we have explored algorithms for cell range extension (CRE) in HetNet scenarios [24], [25]. This can be considered a power domain approach as it extends the range of a low power cell by adding a bias. Approaches 1-3 are generally considered semi-static SON coordination techniques where the goal is to provide the radio scheduler at each cell with information relating to possible interference from neighboring cells. Information exchange occurs across the X2 [1] interface and the time scale of change is in tens of seconds or more. The CoMP approach involves a set of interference minimization strategies, possibly using hybrid combinations of 1-3, across set of geographically separated transmission points (called beams in this work), coordinated by a central scheduler at TTI scale (Transmission Time Interval) [18], [26].

Frequency domain approaches to ICIC involve allocating sub-bands (a collection of one or more PRBs, see Figure 1) to inner and outer cell boundaries. The key objective is to apply orthogonal non-overlapping sub-bands at cell and beam boundaries [27], [28], while applying full frequency reuse at the cell center. Time-domain approaches involve one or more cooperating cells (or equivalently radio base stations) coordinating spectrum within agreed time slots. A detailed survey of time-domain ICIC techniques can be found in [19]. One, standardized, approach to time-domain ICIC is *almost blank subframes* (ABS). The central idea behind this approach is to coordinate in time all used resources in the in the frequency-domain on a subframe (1ms) basis. In this approach one or more cooperating cells (or equivalently radio base stations) will cease sending on all frequency resources associated to a sector or beam for the duration of one or more subframes, leaving exclusive access to other prioritized cells. In LTE, it is, in some specific cases, not possible to blank or silence the subframe completely, due to the need to carry cell reference signals (CRS) which are needed for channel quality measurements. In this case the subframe is instead broadcast at very low power and is almost blank. 5G does not have this limitation as CRS are not used in the same way and thus 5G can use completely silent and thus energy efficient subframes.

We have developed a centralized, policy gradient [29] based deep [30] RL approach to CoMP coordination that can schedule PRBs in both time and frequency-domain. The method can work on TTI or aggregated TTI level and combines approaches one and two discussed above. In the frequency-domain, complex reuse patterns can be learned and dynamically applied, while in the time-domain blanking patterns can be learned. We apply this method to reduce interference patterns among PRBs in a cluster of cooperating cells and beams in order to increase the total cell cluster spectral efficiency[2] [31] and related performance metrics. It is also possible to incorporate the power dimension into our method to cover all the main ICIC approaches.

There has been previous work applying RL to the problem of ICIC. In [32], a cooperative multi-agent approach using tabular Q-learning [9] is applied to learn power reduction strategies on a subset of each cell's bandwidth. Bernardo et al. [33], [34] suggest a cooperative multi-agent approach, using policy gradients, for allocation of sub-bands of frequency to a group of cells. However, unlike our method this approach does not include the time-domain aspect and cannot work at CoMP time scales. Simsek et al. [35] propose a distributed, two level RL, tabular Q-learning approach to learning individual CRE within a cluster of base stations, also associating suitable carrier aggregation [36] frequencies to each device, which is a different allocation problem than the one we study in this paper. In [37] a tabular Q-learning RL approach is applied, in a low dimension action space, to the problem of learning ABS ratios between licensed LTE/5G and an unlicensed radio technology like WiFi to reduce interference. In comparison to all of the above, the method described in the paper applies a deep RL policy gradient approach which can capture a very large action space over PRBs available to each user in a cluster of cells.

[1] 5G allows alternative subcarrier spacings and numerologies [20] to which this method also applies.

[2] The number of bits carried in one second per Hertz of spectrum.

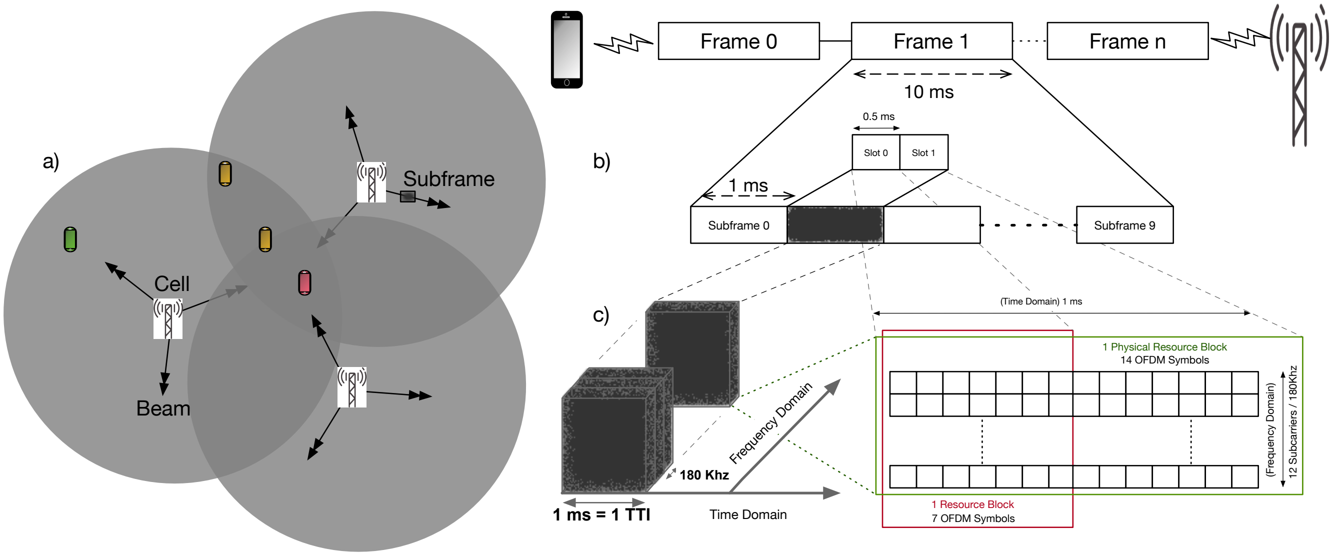


Fig. 1. Radio Environment Overview: (a) shows shows concept of cells and interfering beams. (b) shows time domain structure where a subframe is 1 ms in duration. (c) shows frequency domain structure for each time domain subframe. In 4G basic radio resource (PRB) scheduling is at TTI scale (also 1 ms).

III. Method

A. System Model

The system model consists of a set of cells $\mathcal{C}$, beams $\mathcal{B}$, and available physical resource blocks $\mathcal{R}$ per beam. Each beam is associated with exactly one cell, its origin, and we assume that all PRBs in $\mathcal{R}$ are available in every beam. Each beam also has a configured transmit power $\mathcal{W}$ and gain pattern $\mathcal{G}$, used for fading calculations by a propagation model $\mathcal{P}$ (based on [38], [39], see Table I for details). To this we add a set of users $\mathcal{U}$ to represent service demands. The cells and users are positioned and beams oriented in a geographical grid of size z. This constitutes a *system configuration*

$$\mathcal{S} = \{\mathcal{R}, \mathcal{C}, \mathcal{B}, \mathcal{W}, \mathcal{G}, \mathcal{U}, z\} \tag{1}$$

From $\mathcal{W}$, $\mathcal{G}$ and z we obtain a *nominal* received signal power (RSP) for every user by applying $\mathcal{P}$. We emulate disabling PRBs individually within the beams by decreasing the transmit powers by a fixed ratio g according to a PRB schedule a, and apply a stochastic fast fading emulation $\mathcal{D}$ separately over each element in $\mathcal{R}$, yielding a set of unique RSP samples per PRB, which we treat as *observed* RSPs within emulation episodes. For each PRB, approximate signal-to-noise-and-interference-ratios (SINR) can then be calculated by summing over $\mathcal{R} \times \mathcal{B} \times \mathcal{U}$ and some background noise. For the purpose of our reinforcement learning framework, we designate the resulting SINRs the *observable* states of the modelled system. Finally, a radio resource management scheduling [26] algorithm $\mathcal{M}$ is applied to allocate the non-disabled radio resources to users. How this is done for different combinations of performance objectives is detailed in Section III-E.

This chain of operations, as specified in Equation 2, is applied for each TTI and results in a *realised* spectral efficiency e of shape $|\mathcal{R}| \times |\mathcal{B}| \times |\mathcal{U}|$ and an updated system state of which the observable part is s. This output is together with a chosen PRB schedule a, the input to the RL Framework, as detailed in Section III-C.

$$\{s, e\} \xleftarrow{\text{tti}} \{\mathcal{P}, \mathcal{D}, \mathcal{S}(a), \mathcal{M}\} \tag{2}$$

B. System Model Parameters and Assumptions

The system model replicates key features of a radio network over a geographical grid. Users are placed to represent service demand over which system performance can be measured. Currently the demand model is the simplest possible, known as downlink full buffer (FB) or infinite demand. The fading component $\mathcal{D}$ emulates diffraction and Doppler type effects. Fading state evolves independently per PRB, and is overlaid per user fading based on distance and angles in $\mathcal{S}$. Several types of statistical fading models [40], e.g. Raleigh, Ricean and Weibull are included, but most reported experiments use the Weibull model which also includes a random location specific fading severity, and is auto-correlated over time.

The transferred bits are re-scaled to spectral efficiency and summed over $\mathcal{R} \times \mathcal{B}$ to produce per user statistics, and averaged over $\mathcal{R} \times \mathcal{B} \times \mathcal{U}$ to obtain a system spectral efficiency (SSE) [41] in b/s/Hz. These metrics are mapped to a reward, as described in Section III-E. In our model, without loss of generality, we do not assume any advanced antenna configurations and our maximum SSE (at best SINR) is 4.48 b/s/Hz. The PRB schedule a constitutes an on/off schedule over $\mathcal{R} \times \mathcal{B}$ which can be calculated per TTI or aggregated over TTI intervals. Table I gives further details of the submodels employed. In simulation mode, UEs report on RSP and a channel quality indication (CQI), which is SINR in our model, on all available PRBs. Here we assume that this information is known by the scheduling agent for every beam and PRB.

TABLE I
KEY SYSTEM MODEL PARAMETERS

Parameter	$Value\|Range$	$Note$	$Ref.$
$\mathcal{S}$	System model	See equations 1 and 2	
$\mathcal{P}$	COST 231 Hata	Transmitter height: 10 m	[39]
	$Freq. = 2140\,\mathrm{Mhz}$	Receiver height: 1 m	
$\mathcal{G}$	ITU-R M.2135-1	70° def. horiz. beam width	[38]
$\mathcal{W}$	beam power in dBm	Values used $\{30, 42\}$	
g	Silent PRB	$-30\,\mathrm{dB}$	
$\|\mathcal{R}\|$	1 ms x 180KHz	System values used $\{6, 20\}$	
$\|\mathcal{U}\|$	User devices	Typical values $\{9, 10, 50\}$	
$\mathcal{D}$	standard deviation	$\sim 3\,\mathrm{dB}$	
$\mathcal{M}$	MT or PF	Full Buffer	[26]

In system deployment, this data would be approximated by sampling sub-carrier CQI reports from UEs (see section III-F).

C. Reinforcement Learning Framework

The key parts of a reinforcement learning (RL) framework are an agent, following a policy π, which interacts with an environment through an action a upon which the environment will update its internal state and produce a reward r and a new (observable) state s. The policy $\pi(a|s)$, applied to a state s, is the probability of choosing the action a in state s and the objective of an RL agent is to learn the best a in a given state s for a specific optimization objective. Algorithm 1 outlines the various steps used to integrate this RL approach with our system model. For a short but compact overview of basic RL theory we suggest reviewing this tutorial [42].

```
1  Initialize:
2      Weights of parameterized policy network $\theta$
3      Learning rate $\alpha$; Initial state $s_0$; Trajectory: $\tau$; $b \leftarrow 1$
4  foreach episode: e in E do
5      $s \leftarrow s_0$
6      foreach TTI: $t \in T$ do
7          Sample $a \leftarrow \pi_\theta(s)$
8          Apply system model $\{s, e\} \leftarrow \{\mathcal{P}, \mathcal{D}, \mathcal{S}(a), \mathcal{M}\}$
9          Derive reward $r \leftarrow$ Targ($e$)
10         Record trajectory: $\tau_{\mathbf{e,t}} \leftarrow \{s, a, r\}$
11     end
12     ▷ Update policy every B episodes
13     if $b \mod B = 0$ then
14         $\nabla_\theta \mathcal{J}(\pi_\theta) \leftarrow 0$
15         foreach step i in $\tau$ do
16             $\Delta\theta_i$ = -$\nabla_\theta \log \pi_\theta(a_i|s_i) A^{\pi_\theta}(s_i, a_i)$
17             $\nabla_\theta \mathcal{J}(\pi_\theta) \leftarrow \nabla_\theta \mathcal{J}(\pi_\theta)$ + $\Delta\theta_i$
18         end
19         $\nabla_\theta \mathcal{J}(\pi_\theta) \leftarrow mean(\nabla_\theta \mathcal{J}(\pi_\theta))$
20         ▷ Update policy parameters through
           back-propagation and reset $\tau$
21         $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{J}(\pi_\theta)$
22         Reset $\tau$
23     end
24     $b \leftarrow b + 1$;
25 end
```

Algorithm 1: RL Framework

We use a policy π parameterized by θ, which is implemented (see III-D) as a neural network [43]. The algorithm is run for E episodes, and each episode, a fixed number of time-steps (TTIs), during which actions a are sampled from the current policy $\pi(a \mid s; \theta)$. Each sampled action a is used as input to the system model after which the remaining stages of equation 2 are applied (Line 8).

Each system model iteration produce an updated (observable) state s and a (realised) system efficiency e, both of shape $|\mathcal{R}| \times |\mathcal{B}| \times |\mathcal{U}|$. From e we derive the SSE and per user statistics needed to calculate the reward r. The state s consist of per user SINR or CQI (channel quality indicator) values generated by the system model as a response to the agent's previous action. The derivation of r from e is detailed in Section III-E. We use batching to improve generalization across a number of episodes. A batch is created every B episodes, and the policy is updated using complete TTI trajectories τ. Updates takes place using a policy gradient approach (Lines 15 to 21).

The objective function $\mathcal{J}(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[r(\tau)]$ of an RL framework maximizes the expected reward r, and the policy gradient theorem [44] ensures that the gradient of $\mathcal{J}(\pi_\theta)$ can be can be incrementally calculated across the trajectory τ as done in Lines 16-19[3]. By maximizing the objective over τ in a batch of B episodes and updating the neural network weights θ according to the observed gradients through back-propagation with a suitable learning rate α, the policies π_θ gradually becomes better at selecting actions that maximize reward. The advantage function $A^{\pi_\theta}(s, a)$ encodes the estimation of target reward derived in Line 9, and we can employ several variants as described in Section III-E.

D. Policy Network Architecture

We represent the policy π of Section III-C as a function from state s to the *joint* probability $P(a \mid s)$ of choosing action a in s. Depending on the distribution we assume for P and how we chose to represent a, the structure of neural networks used to represent π can take a few different forms. The simplest one uses a *sigmoid* [43] head over the resources in $\mathcal{R} \times \mathcal{B}$ and the output is interpreted as the probabilities of turning on or off individual PRBs within each beam. A network of this type is illustrated in Figure 2. The actions for this case are sampled as Bernoulli (0/1) outcomes, giving $2^{|\mathcal{R}| \times |\mathcal{B}|}$ possible actions, which can be input directly into the system model to compute the state update, and the per TTI spectral efficiency.

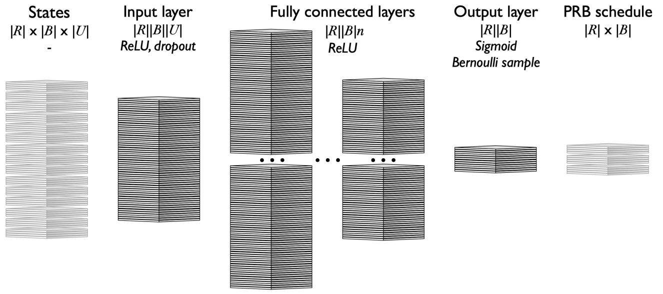

States
$|\mathcal{R}| \times |\mathcal{B}| \times |\mathcal{U}|$
-
Input layer
$|\mathcal{R}||\mathcal{B}||\mathcal{U}|$
ReLU, dropout
Fully connected layers
$|\mathcal{R}||\mathcal{B}|n$
ReLU
Output layer
$|\mathcal{R}||\mathcal{B}|$
Sigmoid
Bernoulli sample
PRB schedule
$|\mathcal{R}| \times |\mathcal{B}|$

Fig. 2. Simulator states, actions and policy network shapes for sigmoid output, where $|\mathcal{R}|$, $|\mathcal{B}|$ and $|\mathcal{U}|$ are the number of PRBs, beams and users respectively.

[3] The $mean$ on Line 19 is calculated per batch, as per Section III-E.

A conceptually more sophisticated model exploits the structure of the dependencies between the individual probabilities produced by π. We can explicitly represent any *combination* of on/off decisions as a separate outcome, and interpret the network head as a single categorical variable using a Softmax [43] operation on the network head. The size of the network output layer grows *very* quickly $\mathcal{O}(2^n)$ with n binary outcomes, which limits the scalability of this model. To reduce the impact of this, we exploit the fact that the PRBs are, by design, largely independent *within* the beams. Under this assumption, we design the network to operate on the input state of the beams and users of a *single* PRB. If the episodes contain more than one PRB, we slice the system state and network head so that PRBs become additional batch samples. This reduces the number of outcomes we need to consider radically, so that it is e.g. entirely realistic to train networks for 6-9 3-sector cells with an arbitrary number of PRBs. Figure 3 illustrates an instance of such a model. Since the same type of slicing may be applied to the sigmoid model, the Softmax model remains more demanding on system resources, however. Section IV-1, explores differences between these models further.

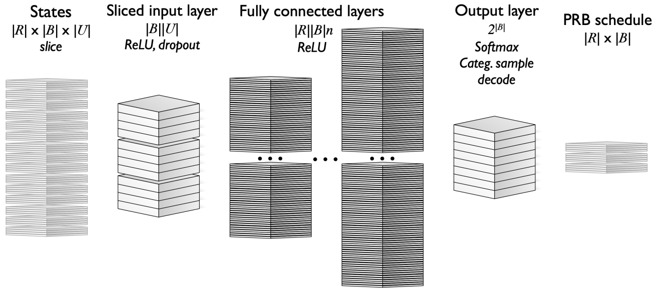

States
$|\mathcal{R}| \times |\mathcal{B}| \times |\mathcal{U}|$
slice
Sliced input layer
$|\mathcal{B}||\mathcal{U}|$
ReLU, dropout
Fully connected layers
$|\mathcal{R}||\mathcal{B}|n$
ReLU
Output layer
$2^{|\mathcal{B}|}$
Softmax
Categ. sample
decode
PRB schedule
$|\mathcal{R}| \times |\mathcal{B}|$

Fig. 3. Sliced policy network shapes with softmax head, where $|\mathcal{R}|$, $|\mathcal{B}|$ and $|\mathcal{U}|$ as in Figure 2. Separate PRBs within the episodes are sliced as additional batch samples. Note exponential size of output layer.

E. Reward Targets and Advantage functions

Several reward functions (Algorithm 1, line 9) can be derived from the spectral efficiency e produced by the system model. The simplest is to average e over $|\mathcal{R}| \times |\mathcal{B}| \times |\mathcal{U}|$ to produce a single per TTI SSE as reward. This corresponds to a maximum throughput (MT) objective.

To take fairness over users into account, we maintain per user throughput running averages T_u . The statistics for these are obtained from e by summing over $|\mathcal{R}| \times |\mathcal{B}|$. From T_u we obtain a proportional fair (PF) reward r by averaging over ${}^{e}\!/\!_{T_u^w}$ where w is a weight assigned to the fairness. Setting $w = 0$ recovers the MT objective, but larger w increases the importance of the fairness in the resulting reward, rapidly approaching a BET (blind equal throughput) [26] objective. Apart from its use in the RL-framework, the reward is also used by the allocation policy $\mathcal{M}$ of the system model. Currently, $\mathcal{M}$ allocates the resource to the user giving the highest reward, but other policies are possible.

Several advantage functions $A^{\pi_\theta}(s, a)$ can be applied in our RL framework. For most of our experiments, we have used REINFORCE [9] with baseline for which $A^{\pi_\theta} = (\tau_r - \bar{r})$, where τ_r, in vector representation, is rewards per TTI for one batch and $\bar{r}$ is a running average reward over a number of batches. The alternative form $(r_i - \bar{r})$ can be used to align with the representation in algorithm 1. From the target, we derive the policy gradient per TTI used by the RL framework as in algorithm 1 line 16. Alternate target functions are possible, and we also tried an *actor-critic* advantage, where a separate value function is trained to predict the reward for a given system state and replaces the baseline in the target return. In our experiments we did not see any obvious advantage of this method.

F. Policy Transfer and Continuous Learning

The RL paradigm facilitates learning optimization policies through reinforced data observations. A policy can be used in, or in place of, an algorithm to achieve some overall system goal. An agent is the general term used to represent a software entity responsible for the policy training process as described in section III-C. One crucial aspect is how a trained policy can be used in an operational system. In our method we propose an initial, agent based, training phase in an offline simulation environment after which the policy is transferred to a RAN system agent framework (see figure 4). We represent the policy as a neural network (see section III-D), implemented as a structured set of weights, which is very portable between simulation and operational system. Using an offline training environment enables access to rich state data without system implementation or resource constraints. Model assumptions made during offline training will, however, change over time and thus we see the need to do continuous, sample based training of policies in an operating RAN context. Access to state data is, in general, limited and resource constrained in a running systems. The RL framework and associated agents will need to collect and aggregate sparse samples from underlying management and observability channels to feed a continuous training process. From a RAN distribution perspective, each RL Framework instance works with a cluster of cells through CoMP coordination mechanisms. We foresee the need for agent coordination strategies between clusters but leave this as future work.

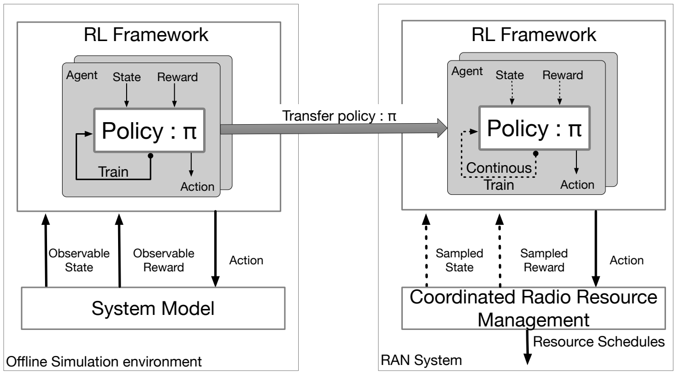
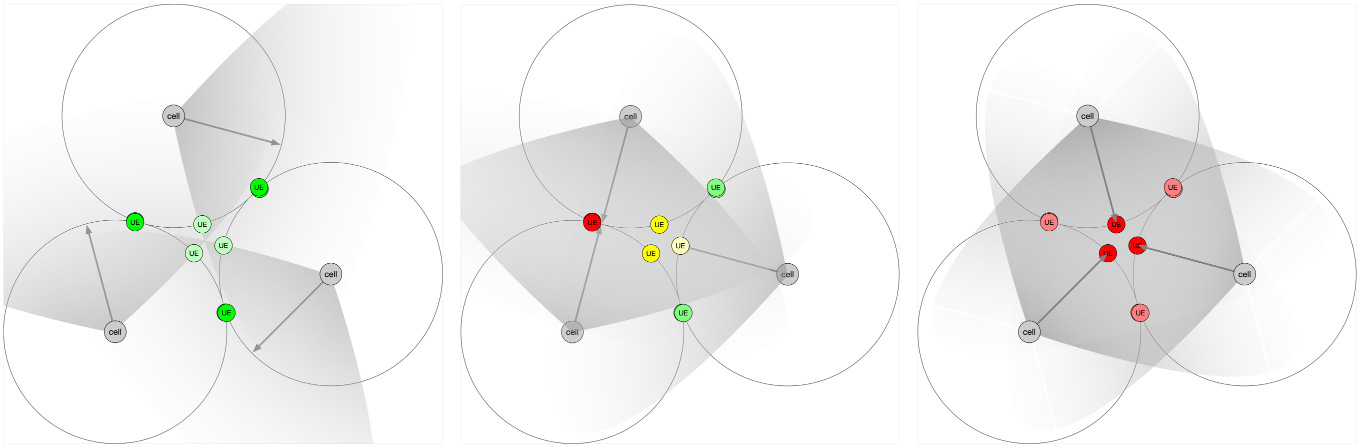


Fig. 4. Policy Transfer.



(a) minimal interference (b) intermediate interference (c) maximum interference

Fig. 5. Three one beam sites. Beam angles for three interference scenarios.

IV. Empirical results

To evaluate our RL method we implemented a highly efficient system model simulation using vectorized Python numpy [45] data structures. The RL framework and related neural network structures, including back-propagation mechanisms, were implemented with PyTorch [46]. The training episodes were run on a Linux infrastructure using powerful GPUs to speed up the process. We performed three different experiment types, detailed in the following sections.

1) Experiment A - Engineered Cell Configurations: To examine the ability of the proposed method to produce schedules under severe interference, we designed scenarios where all users have fixed positions in the central region between three equidistant single beam cells. The angles of the beams are shifted between minimal and maximum interference, as illustrated in Figure 5. We compare the results of training scheduling agents using Softmax (Smx) and Sigmoid (Sig) network architectures (see Section III-D) against two binomial policies. Table II, compares system spectral efficiency (SSE), fairness (Fair.) and proportion of resources scheduled (Ruse) for two objectives, maximum throughput (MT) and proportional fair PF with a weight of 0.2, and using a batch size of $B = 10$, $T = 100$ TTIs per episode, and a total no. of episodes 9000.

Binomial policies take no account of correlating individual on/off variables, only turn them on with a probability according to the distribution parameter p. The measured reuse of a binomial policy will always converge to p. E.g. for binomial $p = 1$, *all* resources will be active while $p = 0.5$ implies a uniformly random policy. For these experiment, two binomial probabilities are chosen for each comparison: One where *all* resources are constantly on (1), and another one (Bin) with p chosen as the resource reuse of the RL trained schedulers. In this way we can see if the trained scheduler just mimics the corresponding binomial policy, or actually improves on it, which is crucial in high interference scenarios.

Inspecting Table II, for the minimal interference case (Fig 5a), all the schedulers show identical results. The binomial-1 scheduler is optimal for this case, since interference is insignificant, and it is encouraging that both trained schedulers produce equivalent results.

For the more severely interfered scenarios (Figures 5b and 5c), both trained schedulers significantly outperform binomial policies in terms of SSE and when compared to binomial for $p = 1$, this holds true at a significantly lower effective reuse, and hence energy expenditure. This implies that the employed interleaving patterns efficiently exploit the variations in SINR to choose the best resources to turn on and off in each individual state. The SSE value 1.56 for the maximum interference case is very close to one third of the theoretical maximum (4.48) under our propagation model, which is the best we could expect in this case.

For the PF case, we see that the SSE goes down as the fairness is improved for all cases, but more so for the binomial policies than the trained ones. For the maximum interference case, the gain from using the trained schedulers is especially significant. The binomial policies, while producing very high fairness, do so at a significantly higher cost in terms of SSE. The trained schedulers apparently avoid producing interference even when supplying less well placed users, a result we consider very significant. Finally we observe that, at least for these comparatively simple examples the Sigmoid appears to reproduce the the results of the Softmax scheduler almost perfectly. If this holds up for more complex scenarios, it is good news, since (a similarly sliced) Sigmoid architecture would be less demanding on memory and compute resources for training.

Figure 6 show convergence plots for key metrics during training of the Softmax scheduler for the maximum interference case. Observe that the reuse initially rises as this improves the SE for well placed users. It also reduces it for the badly placed ones, but this is not immediately captured by the learning policy. Eventually the algorithm learns how to

TABLE II
EXPERIMENT A - ENGINEERED CELL CONFIGURATIONS RESULTS

$\mathcal{M}$	Metric	minimal interference				intermediate interference				maximum interference			
		Smx	Sig	1	Bin	Smx	Sig	1	Bin	Smx	Sig	1	Bin
MT	SSE	4.48	4.48	4.48	4.48	3.04	2.99	2.43	2.18	1.56	1.56	1.07	1.04
	Fair.	0.43	0.43	0.43	0.43	0.42	0.39	0.42	0.52	0.19	0.20	0.90	0.37
	Ruse	1	1	1	1	$2/3$	$2/3$	1	$2/3$	$1/3$	$1/3$	1	$1/3$
PF	SSE	2.89	2.89	2.89	2.89	2.07	2.07	1.76	1.68	1.35	1.35	0.93	0.93
	Fair.	0.61	0.61	0.61	0.61	0.62	0.61	0.59	0.80	0.97	0.97	0.96	0.98
	Ruse	1	1	1	1	$2/3$	$2/3$	1	$2/3$	$2/3$	$2/3$	1	$2/3$
System Config.		$\lvert\mathcal{C}\rvert = 1,\ \lvert\mathcal{B}\rvert = 3,\ \lvert\mathcal{R}\rvert = 6,\ \lvert\mathcal{U}\rvert = 9,\ z = 1\,\mathrm{km}^2,\ \mathcal{W} = 30\,\mathrm{dBm},\ \mathcal{G} = 70^\circ$											

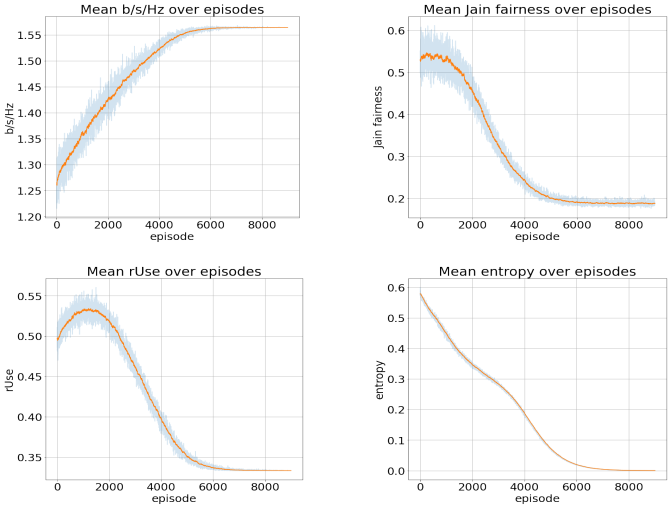


Fig. 6. Engineered configurations: SSE, fairness, reuse factor and entropy training convergence for the Softmax MT scheduler. Config. as in Table II.

disable and interleave PRB usage in order to reduce overall interference and increase SSE. Note also that the policy entropy decrease is not exactly smooth and there is some action space exploration going on between the 2000:th and 4000:th iteration. The Sigmoid scheduler behaves very similarly.

2) Experiment B: In this group of experiments, locations of a specified number of cells and users are randomly generated, then fixed for the duration of training, according to a given seed to allow reproducibility. This gives a random intercell distance with unplanned, but uniformly distributed, beam direction, and random user placement. The *ad hoc* and possibly temporary nature of the configuration makes standard cell planning very difficult. For the purpose of testing our method we generate 100 (randomly selected) of these *ad hoc* network scenarios and apply different system configurations specified as sets 1-3 in Table III. Set-1 uses 3 cells, 3 beams with 6 PRB each, and 10 users in a $1\,\mathrm{km}^2$ grid. For all experiments in B and C, the RL framework uses a batch size $B = 2$ and $T = 250$ time-steps (TTI) per episode, a learning rate $\alpha = 5 \times 10^{-5}$ and the softmax architectural variant (Smx). The RL algorithm trains for $2000 - 8000$ total no. of episodes and the measure of entropy (see Figure 6 for an example entropy curve) can be used for early stopping. Using set-1 configuration, when generating locations as described, we create scenarios with a high probability of users ending up in high interference zones. As for experiment A, we compare the SSE as produced by our RL method (SSE - Smx in Table III) with two binomial policies: 1) Ruse 1 - use all PRB resources; 2) Ruse Bin - match RL algorithm PRB reuse factor. Figure 7-(a) shows the spread of SSE for each of the 100 randomly generated scenarios and shows a visual comparison of our RL method against the SSE produced by the binomial policies. In all almost all cases (except scenario 62 where reuse 1 is marginally better at 0.5%) our method shows significant gains, on average 33% compared against reuse 1 and 48% compared against reuse Bin. Figure 7-(b) shows the % SSE improvement, as a stacked bar chart, across all scenarios for RL, compared against both binomial policies. Possibly even more interesting are the potential energy gains. In this case, we define energy improvement as a term relating number of PRBs not used in those cases where our RL method improves upon reuse 1. For example, when our RL scheduler beats reuse 1 with a reuse factor of 0.7, we define that saving term as $(1 - 0.7)$. In these cases, our RL method can beat reuse 1 in terms of SSE while using significantly fewer PRBs and thus energy in terms of silent PRB, on average 46%, and in one case up to 73%. Figure 7-(c) shows the spread of energy saving across all 100 random network scenarios. As expected, since we use the MT reward objective, fairness suffers. However, as demonstrated in experiment A, it is possible to use the PF objective to balance SSE against fairness.

Set-2 uses the same system parameters as set-1, with one exception; the number of users is increased to 50. There is a higher probability that users fall into cell centers and thus lower probability of MT allocation resources to users in high interference zones. In this case we see lower but still useful SSE gains, on average 5% for reuse 1 and 18% for Bin, across all scenarios. The potential energy gain is still significant for this set, on average 23%, and up to 56%.

Set-3 models a wide area with high power cells by increasing the grid size to $10km^2$, number of cells to 5, PRB to 20 per beam, and the beam output power to $42dBm$. Moreover, in this case the RL method beats both binomial policies by a significant margin giving SSE gains of on average 33% and average energy gains of 49%.

3) Experiment C: In this experiment set (table III set-4), 100 random network scenarios are again generated. The locations of a specified number of cells are randomly gen-

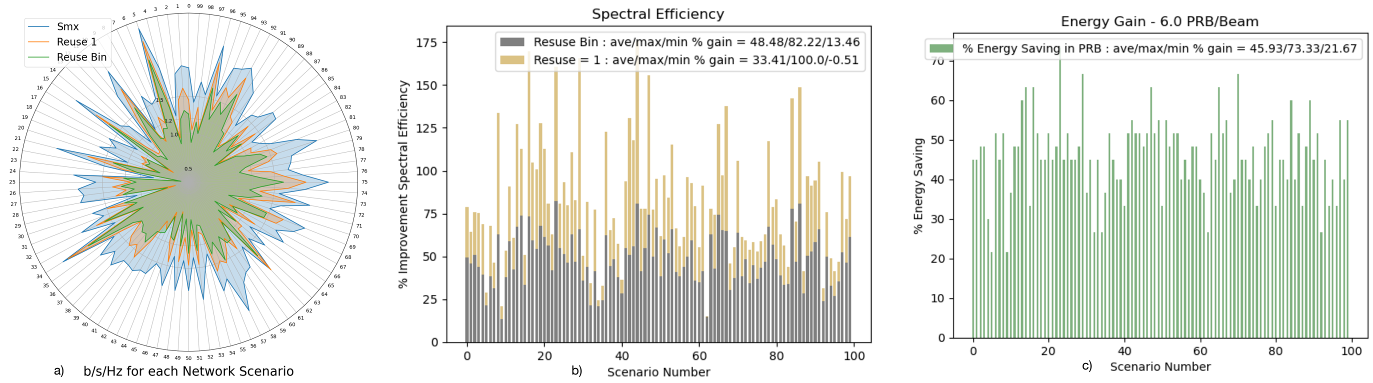


Fig. 7. Experiment B: Set-1 with 3 Cells, 3 Beams, 6 PRBs, 1km, Power = 30dBm. Sub-figure b) is a stacked bar chart.

TABLE III
SUMMARY OF METRICS FOR EXPERIMENTS B AND C USING SOFTMAX ARCHITECTURE

	Experiment B / Set-1 / Fig. 7			Experiment B / Set-2			Experiment B / Set-3			Experiment C / Set-4		
Metric	ave	max	min	ave	max	min	ave	max	min	ave	max	min
SSE - Smx	1.70	2.51	0.82	2.70	3.54	1.59	1.07	1.75	0.60	1.08	1.14	1.02
SSE - Ruse 1	1.32	2.33	0.46	2.60	3.70	1.18	0.83	1.54	0.36	0.87	0.88	0.86
SSE - Ruse Bin	1.17	2.07	0.45	2.32	3.51	1.16	0.69	1.23	0.32	0.79	0.83	0.75
SSE % Gain / Ruse 1	33.41	100.00	-0.51	5.00	34.75	-5.69	32.53	97.22	9.52	24.75	31.40	18.39
SSE % Gain / Ruse Bin	48.48	82.22	13.46	17.77	51.75	0.85	58.78	102.86	33.33	37.25	50.00	25.30
% PRB Energy	45.93	73.33	21.67	23.16	55.50	3.50	48.80	69.00	35.50	51.66	60.00	40.00
Jain Fairness - Smx	0.50	0.69	0.21	0.22	0.30	0.12	0.53	0.74	0.28	0.58	0.64	0.53
Jain Fairness - Ruse 1	0.69	0.89	0.48	0.28	0.45	0.15	0.63	0.86	0.33	0.79	0.82	0.76
Jain Fairness - Ruse Bin	0.57	0.86	0.35	0.24	0.37	0.13	0.61	0.82	0.35	0.68	0.69	0.68
Ruse : Smx	0.54	0.78	0.27	0.77	0.97	0.44	0.51	0.64	0.31	0.48	0.60	0.40
System Configuration	$\|\mathcal{C}\|$=3,$\|\mathcal{B}\|$=9,$\mathcal{M}$=MT $\|\mathcal{R}\|$=6,$\|\mathcal{U}\|$=10 z=1km^2,$\mathcal{W}$=30 dBm			$\|\mathcal{C}\|$=3,$\|\mathcal{B}\|$=9, $\mathcal{M}$=MT $\|\mathcal{R}\|$=20,$\|\mathcal{U}\|$=50 z=1km^2,$\mathcal{W}$=30 dBm			$\|\mathcal{C}\|$=5, $\|\mathcal{B}\|$=15,$\mathcal{M}$=MT $\|\mathcal{R}\|$=20, $\|\mathcal{U}\|$=10 z=10km^2,$\mathcal{W}$=42 dBm			$\|\mathcal{C}\|$=5,$\|\mathcal{B}\|$=15,$\mathcal{M}$=MT $\|\mathcal{R}\|$=20,$\|\mathcal{U}\|$=10 z=1km^2,$\mathcal{W}$=30 dbm		

erated, then fixed for the duration of training, according to a specified seed to allow reproducibility. New user locations are, however, randomly generated for each new episode E. This models a set of *ad hoc* and densely placed cells with extreme mobility among the users. Our RL method again significantly beats both binomial policies with on average 25% SSE improvement over reuse 1, 37% improvement over reuse Bin and 52% energy improvement according to our definition. We consider this result to be especially significant given that random placement/mobility of users each new episode creates a challenging set of scenarios for our RL algorithm.

V. Conclusion and Outlook

Radio spectrum is an incredibly valuable resource and often represents a large monetary investment for cellular operators. Even small fractional improvements in total SSE represent a significant increase in overall system performance. We show that our RL method can autonomously learn sophisticated scheduling patterns and improve SSE by up to 25% in challenging network scenarios. We also show that this SSE improvement can be achieved at significant energy improvement through silencing interfering radio resources. For 5G in particular, lean and energy efficiency radio carriers constitute an important requirement and we see our method being highly applicable, especially in dense *ad hoc* configurations where manual planning is difficult to impossible. We also demonstrate that by selecting an appropriate reward objective, throughput (MT) or fairness (PF), we can elegantly and easily balance SSE with user fairness. This flexible approach can almost certainly be extended to include other objectives like non full buffer traffic, latency awareness or service priority, though we leave these extensions as future work.

One of the truly powerful characteristics of models and policies trained through data, using techniques like RL, is their portability potential. We have proposed a flexible approach where a policy, represented as a neural network, can easily be transferred to a live system for continued contextual training. We see this stepped approach as critical in highly dynamic systems where live contextual patterns can divert significantly from initial assumptions. This is a topic we intend to explore further through multi-agent coordination between clusters of cells.

Acknowledgements

We gratefully acknowledge the help from many colleagues at Ericsson for their valuable comments. Kreuger is funded by Ericsson. Corcoran and Boman are partially funded by the WASP (Wallenberg Autonomous Systems and Software Program) research program. Boman also acknowledges the Swedish Foundation for Strategic Research project *Time-Critical Clouds*.

References

[1] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (EUTRAN); Overall description; Stage 2," ETSI, Tech. Rep. ETSI TS 36.300 V12.5.0, Apr. 2015.

[2] ——, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NR; NR and NG-RAN Overall Description; Stage 2 (Release 15)," 3GPPP, Tech. Rep. 3GPP TS 36.212 V15.1.0, 2018.

[3] S. Hämäläinen, H. Sanneck, and C. Sartori, *LTE self-organising networks:Nnetwork management automation for operational efficiency*. Wiley, 2012.

[4] A. Imran and A. Zoha, "Challenges in 5G: How to empower SON with big data for enabling 5G," *IEEE Netw.*, vol. 28, no. 6, pp. 27–33, 2014.

[5] D. Corcoran, L. Andimeh, A. Ermedahl, P. Kreuger, and C. Schulte, "Data driven selection of DRX for energy efficient 5G RAN," in *2017 13th International Conference on Network and Service Management (CNSM)*, Nov 2017, pp. 1–9.

[6] J. Moysen and L. Giupponi, "From 4G to 5G: Self-organized network management meets machine learning," *Computer Communications*, vol. 129, pp. 248–268, 2018.

[7] N. Vesselinova, R. Steinert, D. F. Perez-Ramirez, and M. Boman, "Learning combinatorial optimization on graphs: A survey with applications to networking," *IEEE Access*, vol. 8, pp. 120 388–120 416, 2020.

[8] D. Corcoran, A. Ermedahl, and C. Granbom, "Artificial Intelligence in RAN – A Software Framework for AI-driven RAN Automation," 2020. [Online]. Available: {https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/artificial-intelligence-in-ran}

[9] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 2nd ed. MIT press, 2018.

[10] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[11] E. Balevi and J. G. Andrews, "Online antenna tuning in heterogeneous cellular networks with deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1113–1124, 2019.

[12] P. Almasan *et al.*, "Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case," *arXiv preprint arXiv:1910.07421*, 2019.

[13] N. Lazic *et al.*, "Data center cooling using model-predictive control," *Proc NeurIPS-18*, 2018, Montreal, pp. 3818-3827.

[14] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[15] F. Vannella *et al.*, "Remote electrical tilt optimization via safe reinforcement learning," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–7.

[16] A. Palm, A. Metzger, and K. Pohl, "Online reinforcement learning for self-adaptive information systems," in *Advanced Information Systems Engineering*, S. Dustdar, E. Yu, C. Salinesi, D. Rieu, and V. Pant, Eds. Cham: Springer International Publishing, 2020, pp. 169–184.

[17] D. Lopez-Perez *et al.*, "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Communications*, vol. 18, no. 3, pp. 22–30, 2011.

[18] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-Advanced Pro and The Road to 5G*, 3rd ed. USA: Academic Press, Inc., 2016.

[19] L. Liu, Y. Zhou, A. V. Vasilakos, L. Tian, and J. Shi, "Time-domain ICIC and optimized designs for 5G and beyond: a survey," *Science China Information Sciences*, vol. 62, no. 2, p. 21302, 2018.

[20] S. Parkvall *et al.*, "NR: The New 5G Radio Access Technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, 2017.

[21] D. Lee *et al.*, "Coordinated multipoint transmission and reception in LTE-advanced: Deployment scenarios and operational challenges," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 148–155, 2012.

[22] Y. L. Lee *et al.*, "Recent advances in radio resource management for heterogeneous LTE/LTE-A networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2142–2180, 2014.

[23] S. Deb *et al.*, "Algorithms for Enhanced Inter-Cell Interference Coordination (eICIC) in LTE HetNets," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 137–150, 2014.

[24] P. Kreuger, O. Görnerup, D. Gillblad, T. Lundborg, D. Corcoran, and A. Ermedahl, "Autonomous load balancing of heterogeneous networks," in *IEEE 81st Vehicular Technology Conference*. IEEE, 2015, pp. 1–5.

[25] P. Kreuger, R. Steinert, O. Görnerup, and D. Gillblad, "Distributed dynamic load balancing with applications in radio access networks," *International Journal of Network Management*, vol. 28, no. 2, 2018.

[26] F. Capozzi *et al.*, "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 678–700, 2013.

[27] M. N. Hindia *et al.*, "Frequency reuse for 4G technologies: A survey," in *Proc ICMSCE 2015*, 2015.

[28] N. Saquib *et al.*, "Fractional frequency reuse for interference management in lte-advanced hetnets," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 113–122, 2013.

[29] V. V. Phansalkar and M. A. L. Thathachar, "Local and global optimization algorithms for generalized learning automata," *Neural Computation*, vol. 7, no. 5, pp. 950–973, 1995.

[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013.

[31] S. Verdú, "Spectral efficiency in the wideband regime," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1319–1343, 2002.

[32] M. Dirani and Z. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in ofdma cellular networks," in *8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2010, pp. 170–176.

[33] F. Bernardo *et al.*, "A novel framework for dynamic spectrum management in multicell ofdma networks based on reinforcement learning," in *IEEE Wireless Communications and Networking*, 2009, pp. 1–6.

[34] ——, "An application of reinforcement learning for efficient spectrum usage in next-generation mobile cellular networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 4, pp. 477–484, 2010.

[35] M. Simsek, M. Bennis, and I. Güvenç, "Learning based frequency- and time-domain inter-cell interference coordination in HetNets," *IEEE Trans on Vehicular Technology*, vol. 64, no. 10, pp. 4589–4602, 2015.

[36] E. Dahlman, S. Parkvall, and J. Sköld, "Carrier aggregation," in *4G LTE-Advanced Pro and The Road to 5G*, 3rd ed., E. Dahlman, S. Parkvall, and J. Sköld, Eds. Academic Press, 2016, pp. 309–330.

[37] Q. Tang, M. Zeng, J. Guo, and Z. Fei, "An almost blank subframe allocation algorithm for 5G new radio in unlicensed bands," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, 2020, pp. 776–781.

[38] M. Telecommunication, "Guidelines for evaluation of radio Mobile Activites of ETRI interface technologies for," no. ITU-R M.2135-1, pp. 1–55, 2009.

[39] Y. Singh, "Comparison of okumura, hata and cost-231 models on the basis of path loss and signal strength," *International Journal of Computer Applications*, vol. 59, pp. 37–41, 2012.

[40] D. Č. Pavlović *et al.*, "Statistics for Ratios of Rayleigh, Rician, Nakagami-m, and Weibull distributed random variables," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[41] M. Toril *et al.*, "Estimating Spectral Efficiency Curves from Connection Traces in a Live LTE Network," *Mobile Information Systems*, vol. 2017, June 2017.

[42] D. Silver, "Deep Reinforcement Learning Tutorial," 2015, [Online; accessed 20-July-2021]. [Online]. Available: {http://videolectures.net/rldm2015_silver_reinforcement_learning/}

[43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[44] R. S. Sutton *et al.*, "Policy gradient methods for reinforcement learning with function approximation," in *Proc NIPS 12*, S. Solla, T. Leen, and K. Müller, Eds. MIT Press, 2000, pp. 1057–1063.

[45] T. Oliphant, "NumPy: A guide to NumPy," USA: Trelgol Publishing, 2006–. [Online]. Available: http://www.numpy.org/

[46] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc NIPS 32*, H. Wallach *et al.*, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.