Optimized Power and Cell Individual Offset for Cellular Load Balancing via Reinforcement Learning

Ghada Alsuhli[†], Karim Banawan[‡], Karim Seddik[†], and Ayman Elezabi[†]
[⋆]*The American University in Cairo,* [‡]*Alexandria University*

***Abstract*—We consider the problem of jointly optimizing the transmission power and cell individual offsets (CIOs) in the downlink of cellular networks using reinforcement learning. To that end, we reformulate the problem as a Markov decision process (MDP). We abstract the cellular network as a state, which comprises of carefully selected key performance indicators (KPIs). We present a novel reward function, namely, the penalized throughput, to reflect the tradeoff between the total throughput of the network and the number of covered users. We employ the twin deep delayed deterministic policy gradient (TD3) technique to learn how to maximize the proposed reward function through the interaction with the cellular network. We assess the proposed technique by simulating an actual cellular network, whose parameters and base station placement are derived from a 4G network operator, using NS-3 and SUMO simulators. Our results show the following: 1) Optimizing one of the controls is significantly inferior to jointly optimizing both controls; 2) our proposed technique achieves 18.4% throughput gain compared with the baseline of fixed transmission power and zero CIOs; 3) there is a tradeoff between the total throughput of the network and the number of covered users.**

I. Introduction

Mobile data traffic is significantly growing due to the rise of smartphone subscriptions, the ubiquitous streaming and video services, and the surge of traffic from social media platforms. Global mobile data traffic is estimated to be 38 exabytes per month in 2020 and is projected to reach 160 exabytes per month in 2025 [1]. To cope with this high traffic volume, considerable network-level optimization should be performed. Cellular network operators strive to enhance the users' experience irrespective of the traffic demands. This entails maximizing the average throughput of the users and minimizing the number of users that are out of coverage.

The indicated optimization problem is challenging for its often contradicting requirements and the absence of accurate statistical models that can describe the tension(s). Specifically, increasing the transmitted power of some base station (an eNB in LTE[1]) may enhance the SINR for its served users. However, cell edge users in neighboring cells will have increased interference. Hence, network-wide, the optimization based on transmit power alone may tend to sacrifice cell edge users in favor of maximizing overall throughput. Another technique of throughput maximization is mobility load management. This can be done by freeing the physical resource blocks (PRBs) of congested cells by forcing edges users to handover to less-congested cells. A typical approach is controlling the cell individual offset (CIO) [2]. The CIO is an offset that artificially alters the handover decision. This may enhance the throughput by allowing the users to enjoy more PRBs compared to their share in the congested cells. Nevertheless, this may result in decreasing the SINR of users handed over, mostly edge users, as the CIO decision only fakes the received signal quality. This in turn suggests a modified throughput reward function to extend the coverage for cell edge users. The complex interplay between the SINR, available PRBs, CIOs, and the transmitted power is challenging to model using classical optimization techniques. This motivates the use of machine learning techniques to dynamically and jointly optimize the power and CIOs of the cells.

LTE and future networks are designed to support self-optimization (SO) functionalities [3]. A significant body of literature is concerned with load balancing via self-tuning of CIO values, e.g., [2], [4]–[9]. Balancing the traffic load between neighboring cells by controlling the coverage area of the cells appears in [10]–[13]. Reshaping the coverage pattern is usually performed by adjusting the transmit power [10], the transmit power and antenna gain [11], or the directional transmit power [12] of the eNB. To the best of our knowledge, the joint optimization problem of power and CIO using reinforcement learning has not been investigated in the literature.

In this work, we propose a joint power levels and CIOs optimization using reinforcement learning (RL). To that end, we recast this optimization-from-experience problem as a Markov decision problem (MDP) to model the interaction between the cellular network (a.k.a., the environment) and the decision-maker (a.k.a., the central agent). The MDP formulation requires a definition of a state space, an action space, and a reward function. The state is a compressed representation of the cellular network. We define the state as a carefully-chosen subset of the key performance indicators (KPIs), which are usually available at the operator side. This enables the seamless application of our techniques to current cellular networks. We propose a novel action space, where both power levels and CIOs are utilized to enhance the user's quality of

[1]It is worth noting that our proposed reinforcement learning technique can be seamlessly applied to 5G networks as well. Our simulations are based on LTE and we refer to LTE throughout the paper since our data comes from a 4G operator.

service. We argue that both controls have distinct advantages. Furthermore, we propose a novel reward function, which we call the penalized throughput, that takes into consideration the total network throughput and the number of uncovered users. The penalty discourages the agent from sacrificing edge-users to maximize the total system throughput.

Based on the aforementioned MDP, we use actor-critic methods [14] to learn how to optimize the power and CIO levels of all cells from experience. More specifically, we employ the twin deep delayed deterministic policy gradient (TD3) [15] to maximize the proposed penalized throughput function. The TD3 technique is a state-of-the-art RL technique that deals with continuous action spaces. In TD3, the critics are neural networks (NNs) for estimating state-action values (a.k.a., the Q-values). Meanwhile, the actor is a separate NN that estimates the optimal action. The actor function is updated in such a way that maximizes the expected estimated Q-value. The critics are updated by fitting a batch of experiences that are learned from the interactions with the cellular network.

We gauge the efficacy of our proposed technique by constructing an accurate simulation suite using the NS-3 and SUMO simulators. We simulated an actual 4G cellular network that is currently operational in the Fifth Settlement neighborhood in Cairo, Egypt. The site data including the base station placement is provided by a major 4G operator in Egypt. Our numerical results show the validity of our claim that using one of the controls (transmitted power or CIO) but not both is strictly sub-optimal with respect to joint optimization in terms of the throughput. Thus, our proposed technique outperforms its counterpart in [2] by 11%. Furthermore, our technique results in significant gains in terms of the channel quality indicators (CQIs) and the network coverage. Finally, our proposed penalized throughput effects a tradeoff between the overall throughput and the average number of covered users that can be controlled.

II. System Model

We consider the downlink (DL) of a cellular system with N eNBs. The nth eNB sends its downlink transmission with a power level $P_n \in [P_{\min}, P_{\max}]$ dBm. The cellular system serves K mobile user equipment (UEs). Each UE measures the reference signal received power (RSRP) on a regular basis and connects to the cell that results in the best-received signal quality [16]. Thus, at $t = 0, 1, 2, \cdots$, there are $K_n(t)$ UEs connected to the nth eNB such that $\sum_{n=1}^{N} K_n(t) \leq K$. The kth UE moves with a velocity v_k along a mobility pattern that is unknown to any of the eNBs. The kth UE periodically reports the CQI, ϕ_k to the connected eNB. The CQI is a discrete measure of the quality of the channel perceived by the UE that takes a number from the set $\{0, 1, \cdots, 15\}$. When $\phi_k = 0$, the kth UE is out of coverage, while a higher value of ϕ_k corresponds to higher channel quality and hence results in a better modulation and coding scheme (MCS) assignment. The kth UE requests a minimum data rate of ρ_k bits/s.

The connected eNB assigns B_k PRBs to the kth UE as:

$$B_k = \left\lceil \frac{\rho_k}{g(\phi_k, M_{n,k})\Delta} \right\rceil \tag{1}$$

where $g(\phi_k, M_{n,k})$ is the spectral efficiency achieved by the scheduler with a UE having a CQI of ϕ_k and an antenna configuration of $M_{n,k}$, and $\Delta = 180$ KHz, which is the bandwidth of a resource block in LTE. The total PRBs needed to serve the UEs associated with the nth eNB at time t is given by $T_n(t) = \sum_{k=1}^{K_n(t)} B_k$. Denote the total available PRBs at the nth eNB by Σ_n.

Furthermore, we assume that there exists a central agent[2] that can monitor all the network-level KPIs and aims at enhancing the user's experience. Aside from controlling the *actual* power P_n, the agent can control CIOs. The relative CIO of cell i with respect to cell j is denoted by $\theta_{i\to j} \in [\theta_{\min}, \theta_{\max}]$ dB, and is defined as the offset power in dB that makes the RSRP of the ith cell *appears* stronger than the RSRP of the jth cell. Controlling the power levels $(P_n : n = 1, 2, \cdots, N)$ and the CIOs $(\theta_{i\to j} : i \neq j, i, j = 1, 2, \cdots, N)$ can trigger the handover procedure for the edge UEs. More specifically, a UE which is served by the ith cell may handover to the jth cell if the following condition holds [18]:

$$Z_j + \theta_{j\to i} > H + Z_i + \theta_{i\to j}, \tag{2}$$

where Z_i, Z_j are the RSRP from cells i, j, respectively, and H is the hysteresis value that minimizes the ping-pong handover scenarios due to small scale fading effects. By controlling the handover procedure, the traffic load of the network is balanced across the cells. The agent chooses a policy such that the total throughput and the coverage of the network are simultaneously maximized in the long run according to some performance metric as we will formally describe next.

III. Reinforcement Learning Framework

In this section, we recast the aforementioned problem as an MDP. MDPs [19] describe the interaction between an *agent*, which is in our case the power levels and the CIOs controller, and an *environment*, which is the whole cellular system including all eNBs and UEs. At time $t = 0, 1, 2, \cdots$, the agent observes a representation of the environment in the form of a *state* $S(t)$, which belongs to the state space $\mathcal{S}$. The agent takes an *action* $A(t)$, which belongs to the action space $\mathcal{A}$. This causes the environment to transition from the state $S(t)$ to the state $S(t+1)$. The effect of the action $A(t)$ is measured through a *reward function*, $\mathcal{R}(t+1) \in \mathbb{R}$. To completely recast the problem as an MDP, we need to define $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{R}(\cdot)$ as we will show next.

A. Selection of the State Space

To construct an observable abstraction of the environment, we use a subset of the network-level KPIs as in [2], [20]. Since the KPIs are periodically reported by the eNBs, there is

[2]The 3GPP specifies an architecture for centralized self-optimizing functionality, which is intended for maintenance and optimization of network coverage and capacity by automating these functions [17].

no added overhead at communicating the state to the agent. In this work, the state comprises of the following components: First, the resource block utilization (RBU) vector, $\mathbf{U}(t) = [U_1(t) \quad U_2(t) \quad \cdots \quad U_N(t)] \in [0,1]^N$, where $U_n(t) = \frac{T_n(t)}{\Sigma_n}$ is the RBU of the nth eNB at time t. The RBU reflects the load level of each cell. Second, the DL throughput vector, $\mathbf{R}(t) = [R_1(t) \quad R_2(t) \quad \cdots \quad R_N(t)] \in \mathbb{R}_+^N$, where $R_n(t)$ is the total DL throughput at the nth cell at time t. Third, the connectivity vector $\mathbf{K}(t) = [K_1(t) \quad K_2(t) \quad \cdots \quad K_N(t)] \in \mathbb{N}^N$, where $K_n(t)$ is the number of active UEs that are connected to the nth eNB. This KPI shows the effect of the handover procedure. Furthermore, the average user experience at the nth cell is dependent on $K_n(t)$ as the average throughput per user, $\bar{R}_n(t)$ is given by $\bar{R}_n(t) = \frac{R_n(t)}{K_n(t)}$. Finally, the low-rate MCS penetration matrix $\mathbf{M}(t) \in [0,1]^{N \times \tau}$, where τ is the number of low-rate MCS combinations[3]. The element (i,j) of the matrix $\mathbf{M}(t)$ corresponds to the ratio of users in the nth cell that employs the jth MCS. The matrix $\mathbf{M}(t)$ gives the distribution of relative channel qualities at each cell.

Now, we are ready to define our state $S(t)$, which is simply the concatenation of all four components as:

$$S(t) = [\mathbf{U}(t)^T \quad \mathbf{R}(t)^T \quad \mathbf{K}(t)^T \quad \text{vec}(\mathbf{M}(t))^T]^T \tag{3}$$

where vec$(\cdot)$ is the vectorization function. The connectivity and the throughput vectors are normalized to avoid having dominant features during the weight-learning process.

B. Selection of the Action Space: CIO and Transmitted Power

In this work, the central agent has two controls, namely, the power levels and the CIOs. More specifically, the agent selects the action $A(t)$ which is a vector of $\frac{N(N+1)}{2}$ dimensionality,

$$A(t) = [\mathbf{P}^T \quad \boldsymbol{\theta}^T]^T \tag{4}$$

where $\mathbf{P} = [P_1 \quad P_2 \quad \cdots \quad P_N] \in [P_{\min}, P_{\max}]^N$ is the eNB power level vector, and $\boldsymbol{\theta} = [\theta_{i \to j}(t) : i \neq j, \; i,j = 1, \cdots, N] \in [\theta_{\min}, \theta_{\max}]^{N(N-1)/2}$ is the relative CIO vector[4]. Define $\alpha_L = [P_{\min} \cdot \mathbf{1}_N \quad \theta_{\min} \cdot \mathbf{1}_{N(N-1)/2}]$ and $\alpha_H = [P_{\max} \cdot \mathbf{1}_N \quad \theta_{\max} \cdot \mathbf{1}_{N(N-1)/2}]$ to be the limits of $\mathcal{A}$.

We argue that both CIOs and power levels are strictly beneficial as steering actions of the agent. To see that, we note that the power level control corresponds to an *actual* effect on the channel quality at non-edge UEs and the interference faced by the edge UEs. More specifically, as the power level of the nth eNB P_n increases, the channel quality of the non-edge UEs at the nth cell increases while the inter-cell interference faced by edge UEs of the neighboring cell increases as well. This is not the case if the CIO control is used as the CIO *artificially* triggers the handover procedure without changing the power level. Furthermore, the power level control is not sufficient to solve our problem. This is due the fact that the power level P_n controls all the edges of the nth cell simultaneously, while the CIO $\theta_{n \to m}$ can be tailored such that it controls only the cell edge that is common with the mth cell only without affecting the remaining edges. Hence, both techniques have complementary advantages that ultimately result in the superior performance gain.

C. Selection of the Reward Function: Penalized Throughput

To assess the performance of a proposed policy in an MDP, one should formulate the goal of the system in terms of a reward function $\mathcal{R}(\cdot)$. The central agent in MDP implements a stochastic policy[5], π, where $\pi(a|s)$ is the probability that the agent performs an action $A(t) = a$ given it was in a state $S(t) = s$. The central agent aims at maximizing the expected long-term sum discounted reward function [19], i.e.,

$$\max_{\pi} \lim_{L \to \infty} \mathbb{E}_{\pi} \left[\sum_{t=0}^{L} \lambda^t \mathcal{R}(t) \right] \tag{5}$$

where λ corresponds to the discount factor, which signifies how important future expected rewards are to the agent.

The UE desires to be served consistently with the highest possible data rate. One possible reward function to reflect this requirement is the total throughput of the cellular system, i.e.,

$$\mathcal{R}(t) = \sum_{n=1}^{N} \sum_{k_n=1}^{K_n} R_{k_n}(t) \tag{6}$$

where $R_{k_n}(t)$ is the actual measured throughput of the k_nth user at time t. This also reflects the average user's throughput by normalizing by the total number of UEs K.

Now, since the power levels and the relative CIOs are controllable, the agent may opt to choose power levels that effectively shrink the cells' radii or CIOs levels that connect the edge UEs to a cell with poor channel quality. In this case, the agent maximizes the total throughput by only keeping the non-edge UEs that enjoy high MCS and at the same time minimizing the inter-cell interference. Therefore, using total throughput as the sole performance metric is not suitable for representing the user experience as edge users may be out of coverage even if the total throughput is maximized.

In this work, we propose a novel reward function, namely, the penalized throughput. This entails maximizing the average user throughput while minimizing the number of uncovered UEs. More specifically, our reward function is defined as:

$$\mathcal{R}(t) = \sum_{n=1}^{N} \sum_{k_n=1}^{K_n} R_{k_n}(t) - \eta \bar{R}(t) \sum_{k=1}^{K} \mathbb{1}(\phi_k = 0) \tag{7}$$

where $\mathbb{1}(X) = 1$ if the logical condition X is true and 0 otherwise, $\bar{R}(t) = \frac{1}{K} \sum_{n=1}^{N} \sum_{k_n=1}^{K_n} R_{k_n}(t)$ is the average user throughput at time t, and η is a hyperparameter that signifies how important is the coverage metric with respect the total throughput. Our reward function implies that the total

[3]Ideally, we would consider the total number of MCS combinations, which is 29 in LTE. However, to reduce the dimensionality of the state $S(t)$ to ensure stable convergence of the RL, we focus only on a number of low-rate MCSs (e.g., $\tau = 10$). This is because our controlling actions primarily affect the edge users, who are naturally assigned a low-rate MCSs.

[4]Without loss of genrality, we assume that $\theta_{i \to j} = -\theta_{j \to i}$

[5]Generally, the MDP framework allows for the use of stochastic policies. Nevertheless, in this work, we focus only on the special case of deterministic policies, i.e., $p(a^*|s) = 1$ for some $a^* \in \mathcal{A}$.

throughput is decreased by the throughput of the UEs that are out of coverage (assuming that all UEs enjoy the same throughput $\bar{R}(t)$). In practice, the user is considered to be uncovered when the reported CQI by the UE equals zero [18], and hence dropped at the MAC scheduler.

IV. Proposed Reinforcement Learning Technique

We employ an actor-critic RL technique to solve our optimization problem (see Fig. 1). Different from Q-learning [21], the actor-critic methods [14] construct distinct NNs to separately estimate the Q-value and the best possible action based on the observed state. This distinction enables the actor-critic methods to deal with a continuous action space as they do not require to tabulate all possible action values as in deep Q-learning (DQN). Since the number of actions is exponential in the number of eNBs, tabulating action values in DQN as in [2] becomes prohibitive for large cellular networks. The actor function $\mu(s)$ outputs the best action for the state s, while the critic function $Q(s,a)$ evaluates the quality of the (s,a) pair.

In this work, we employ the TD3 technique [15]. Similar to its predecessor the deep deterministic policy gradient (DDPG) [22], the TD3 uses the experienced replay technique, in which a buffer $\mathcal{D}_B$ of size B is used to collect the experiences of the RL agent. More specifically, at every interaction with the environment, the tuple $(S(t), A(t), S(t+1), \mathcal{R}(t+1))$ is stored in the buffer. To update the weights of the NN, a random batch of size $B_m < B$ is drawn from the buffer and used to update the weights. This breaks the potential time correlation between the experiences and ensures better generalizability.

The TD3 technique improves the performance of DDPG by employing a pair of independently trained critic functions instead of one as in the case of DDPG. The TD3 technique chooses the smallest Q-value of the two critics to construct the target network. This leads to less-biased Q-value estimation in addition to decreasing the variance of the estimate due to this underestimation as underestimation errors do not propagate through updates. Additionally, the actor function is updated every T_u time steps, where T_u is a hyper-parameter of the scheme. Delaying the updates result in a more stable Q-value estimation. Finally, TD3 uses a target smoothing regularization technique that adds clipped noise to the target policy before updating the weights. This leads to a smoother Q-value estimation by ensuring that the target fitting is valid within a small neighborhood from the used action. In the sequel, we describe the algorithm in detail.

Our implementation of the TD3 is as follows:

1) *Initialization:* The experience replay buffer $\mathcal{D}_B$ is initially empty. The TD3 uses two NNs as critic functions, $Q(s,a;\mathbf{w}_t^{c_1})$ and $Q(s,a;\mathbf{w}_t^{c_2})$, where $\mathbf{w}_t^{c_i}$, $i=1,2$ is the weight vector of the ith critic function. The TD3 uses a NN for the actor function $\mu(s;\mathbf{w}_t^a)$, where $\mathbf{w}_t^a$ is the NN weight vector of the actor function. We randomly initialize the weights $\mathbf{w}_t^{c_1}$, $\mathbf{w}_t^{c_2}$, $\mathbf{w}_t^a$. Furthermore, we construct target NNs corresponding to the critics and the actor with weight vectors $\bar{\mathbf{w}}_t^{c_1}$, $\bar{\mathbf{w}}_t^{c_2}$, $\bar{\mathbf{w}}_t^a$, respectively.

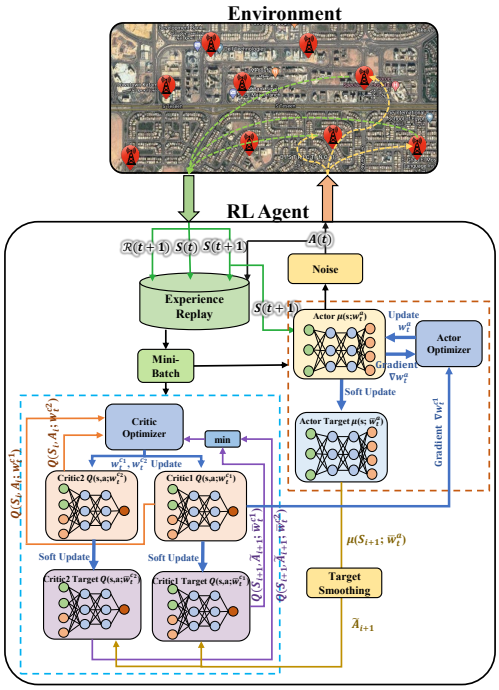


Fig. 1: The proposed load balancing model

Initially, we set these weights to their respective main weights as $\bar{\mathbf{w}}_t^{c_i} \leftarrow \mathbf{w}_t^{c_i}$ for $i=1,2$ and $\bar{\mathbf{w}}_t^a \leftarrow \mathbf{w}_t^a$.

2) *Action Space Exploration:* The agent explores the action space $\mathcal{A}$ by adding an uncorrelated Gaussian noise $\mathcal{N}(0,\sigma^2)$ to the output of the actor function, i.e., assuming that the cellular system at time $t=0,1,\cdots$ is at state $S(t)$, the agent chooses an action $A(t)$ such that:

$$A(t) = \text{clip}(\mu(S(t);\mathbf{w}_t^a) + \epsilon, \alpha_L, \alpha_H) \tag{8}$$

where ϵ is a noise vector, whose kth component $\epsilon(k) \sim \mathcal{N}(0,\sigma_n^2)$; $\text{clip}(x,a,b) = a$ if $x < a$, $\text{clip}(x,a,b) = b$ if $x > b$, and $\text{clip}(x,a,b) = x$ if $a \le x \le b$. The clip function ensures that the exploration is within the action space limits (α_L, α_H). The agent applies the action $A(t)$ and observes the new state $S(t+1)$ and the reward function $\mathcal{R}(t+1)$. The experience $(S(t), A(t), S(t+1), \mathcal{R}(t+1))$ is stored in the buffer $\mathcal{D}_B$.

3) *Critics Update:* Firstly, we randomly draw a batch of size B_m from $\mathcal{D}_B$. For the ith sample of the batch $(S_i, A_i, \mathcal{R}_{i+1}, S_{i+1})$, where $i=1,2,\cdots,B_m$, we use the target actor network to compute the target action,

$$A_{i+1} = \mu(S_{i+1};\bar{\mathbf{w}}_t^a) \tag{9}$$

Then, the smoothed target action $\tilde{A}_{i+1}$ is calculated by adding the clipped noise, such that

$$\tilde{A}_{i+1} = \text{clip}(A_{i+1} + \tilde{\epsilon}, \alpha_L, \alpha_H), \;\; i=1,\cdots,B_m \tag{10}$$

where $\tilde{\epsilon} = \text{clip}(\mathcal{N}(0,\tilde{\sigma}_n^2), -c, c)$ for some maximum value $c > 0$. Secondly, the target function is calculated using the *minimum estimate* of the Q-value from the two

target critics for the perturbed input, i.e.,

$$y_i = \mathcal{R}_{i+1} + \lambda \min_{j=1,2} Q(S_{i+1}, \tilde{A}_{i+1}; \bar{\mathbf{w}}_t^{c_j}) \quad (11)$$

The weights of the two critics are updated by minimizing the mean square error across the batch, i.e., for $j = 1, 2$,

$$\mathbf{w}_{t+1}^{c_j} = \arg\min_{\mathbf{w}_t^{c_j}} \frac{1}{B_m} \sum_{i=1}^{B_m} (y_i - Q(S_i, A_i; \mathbf{w}_t^{c_j}))^2 \quad (12)$$

4) *Actor Update:* TD3 updates the actor function every T_u time steps. To update the actor function, TD3 *maximizes* the expected Q-value function, therefore, the scheme calculates the gradient ascent of the expected Q-value with respect to $\mathbf{w}_t^a$, i.e., TD3 calculates $\nabla_{\mathbf{w}_t^a} \mathbb{E}[Q(s, a; \mathbf{w}_t^c)|s = S_i, a = \mu(S_i; \mathbf{w}_t^a)]$, which can be approximated as:

$$\frac{1}{B_m} \sum_{i=1}^{B_m} \nabla_a Q(s, a; \mathbf{w}_t^c)|_{\substack{s=S_i, \\ a=\mu(S_i;\mathbf{w}_t^a)}} \nabla_{\mathbf{w}_t^a} \mu(s; \mathbf{w}_t^a)|_{s=S_i} \quad (13)$$

This results in new weights $\mathbf{w}_{t+1}^a$.

5) *Target Networks Update:* TD3 uses soft target updates, i.e., the target NNs are updated as a linear combination of new learned weights and old target weights,

$$\bar{\mathbf{w}}_{t+1}^c \leftarrow \gamma \mathbf{w}_{t+1}^c + (1 - \gamma) \bar{\mathbf{w}}_t^c \quad (14)$$

$$\bar{\mathbf{w}}_{t+1}^a \leftarrow \gamma \mathbf{w}_{t+1}^a + (1 - \gamma) \bar{\mathbf{w}}_t^a \quad (15)$$

where γ is the soft update coefficient, which is a hyper-parameter chosen from the interval $[0, 1]$. This constrains the target values to vary slowly and stabilizes the RL.

V. Numerical Results

In this section, the performance of the proposed approach is evaluated through simulations. The simulated cellular network consists of $N = 6$ irregular cells distributed in the area of 900m$\times$1800m extracted from the Fifth Settlement neighborhood in Egypt. There are $K = 40$ UEs with realistic mobility pattern created by the Simulation of Urban Mobility (SUMO) according to the mobility characteristics in [23]. The UEs are either vehicles or pedestrians. The walking speed of the pedestrians ranges between $0 - 3$m/s. All UEs are assumed to have full buffer traffic model, i.e., all UEs are active all the time. This cellular network, which represents the environment of the proposed RL framework, is implemented using LTE-EPC Network Simulator (LENA) [24] module which is included in NS-3 Simulator. The agent is implemented using Python, which is based on the Open AI Gym implementation in [25]. The interface between the NS3-based environment and the agent is implemented via the NS3gym interface [26]. This interface is responsible of applying the agent's action to the environment at each time step. Then, the network is simulated having selected the action in effect. Afterwards, the reward is estimated based on the expression in (7). Finally, the NS3gym interface returns the reward and the environment state back to the agent. Table I presents our simulators' configuration parameters. To show the effectiveness of our RL framework, we compare the performance of the following four control schemes: 1) CIO control: In this scheme, The actions that the agent specify are the relative CIOs between every two neighboring cells; we restrict our CIOs in our simulation setup to be in the range $[-6, 6]$. Whereas, the transmission power remains constant at 32 dBm for all cells in the network[6]. 2) Power control: Here, all CIOs are set to be zeros and the transmission power values are determined by the agent within the range $[32 - 6, 32 + 6]$ dBm. 3) CIO and transmitted power controls: This is our proposed action space. The agent determines the values of the relative CIOs and the transmission power within the ranges $[-6, 6]$ and $[32 - 6, 32 + 6]$, respectively. 4) Baseline scheme: In this scheme, no load management is assumed. The CIOs are set to zeros and the transmission power values are set to be 32dBm for all cells.

	Parameter	Value
TD3	Batch size (B_m)	128
	Policy delay	2 steps
	Layers (N_h, n_i, Activation)	(2, 64$\times$64, ReLu)
	Discount factor	0.99
	Number of episodes	250
Env.	Number of steps/episode	250
	Step time	200 ms

TABLE I: Simulation parameters

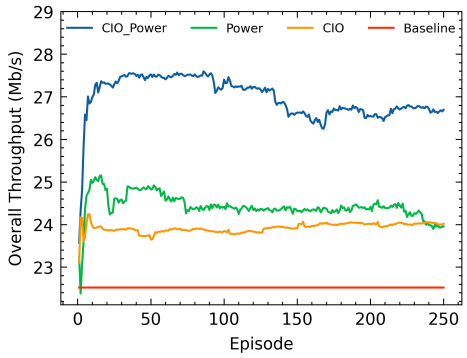


Fig. 2: Average overall throughput during the learning process

The relative performance of the previously mentioned control schemes is shown in Fig. 2, Fig. 3, and Fig. 4. The average overall system throughput, the average CQI, and the average number of covered users are used as quality indicators of the different schemes. These indicators are averaged over 250 steps in each episode and observed for 250 episodes of the learning process. All reported results are obtained by averaging over 10 independent runs to reduce the impact of the exploration randomness on the relative performance of the schemes. In Fig. 2, the proposed power and CIO control

[6]Note that, the mean power level of 32 dBm is a typical operational transmitted power value.

scheme outperforms the CIO control scheme by 11%, the power control scheme by 11.3%, and the baseline scheme by 18.4%, in term of overall throughput after 250 episodes. This is because the adopted scheme combines the advantages of CIO control, of flexible and asymmetric control, and the transmission power control, which allows for better channel quality and better interference management between the cells. These advantages are translated to better average CQI for the proposed scheme in Fig. 3. The worst CQI is associated with the CIO control scheme. This happens because the actual RSRP is counterfeited, by adding the CIO values in equation (2), to trigger the handover of a UE to an underutilized cell with lower channel quality.

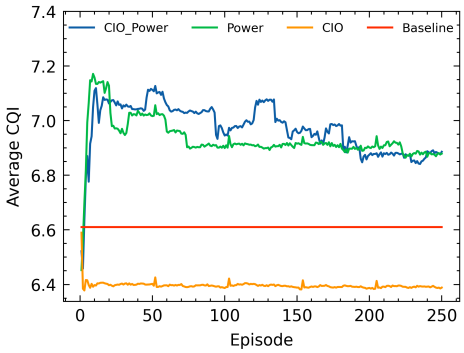


Fig. 3: Average CQI during the learning process

Fig. 4 plots the number of covered users averaged over each learning episode. We observe that none of the schemes attain a number of covered users of 40 UEs, despite the adopted full traffic model. This implies the presence of out of coverage users problem. Fig. 4 shows that the baseline scheme presents near-optimal number of covered users (40 UEs). When the power control is used, decreasing the value of the transmission power of a specific cell without increasing the transmission power of the neighboring cells causes gaps in coverage between these cells. Then, the UEs located in these gaps are uncovered. With the CIO control, the probability of connecting the UEs to a cell with lower CQI, and thus the probability of having more out of coverage users is higher. As a result, this problem is clearer in case of using CIO control. By using both controls, the average number of covered users increases with respect to the CIO control, but still remains inferior to using the transmitted power control only. In summary, The adopted control scheme achieves better overall throughput, better average channel quality indicator, and less out of coverage users problem compared to the CIO control.

Next we investigate the effect of our reward function on the proposed RL framework. We show the radio environment map (i.e., SINR distribution) of the simulated network at end of a specific time step in Fig. 5. The letters A to F represent eNBs sites, while the numbers 1 to 40 correspond to UE locations. The circled UEs are reported as out of coverage users. In Fig. 5, two agents are trained to control both CIOs and tranmsitted power levels with different penalty factors (η). In Fig. 5a, the target of the agent is maximizing the

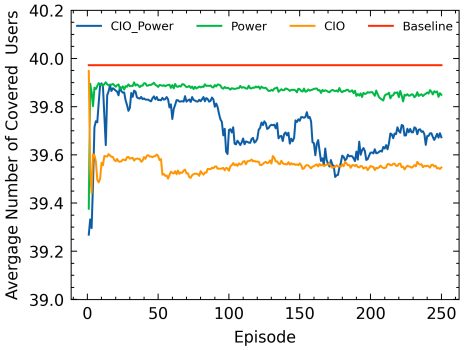


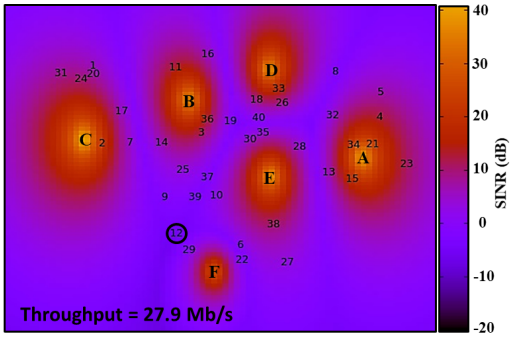
Fig. 4: Average number of covered users during the learning

proposed reward in this paper with $\eta = 2$, i.e. maximizing the throughput while minimizing the number of out of coverage users. In Fig. 5a, the agent decided to select an action that results in 1 uncovered user and 27.9 Mb/s overall throughput for the presented distribution of the users. On the other hand, when the penalty for the out of coverage users is not considered (i.e., $\eta = 0$) in Fig. 5b, the selected action by the agent for the same user distribution causes 6 users to be uncovered; however, we can see that a higher throughput of 30.1Mb/s is achieved. More specifically, the agent intentionally tries to force more out of coverage users as long as this increases the overall throughput. For instance, the agent chose an action that attach UE 27 with cell (A) although the UE is located closer to the coverage of the less utilized cell (F).

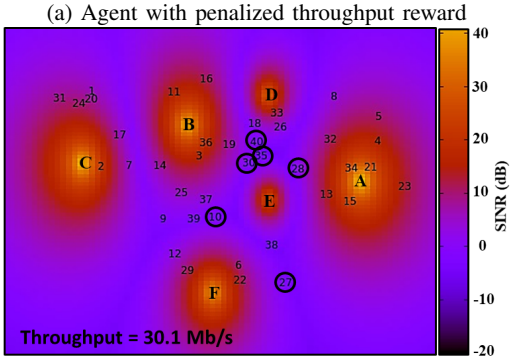
Fig. 6 shows the tradeoff between the overall throughput and the average number of covered users. The two sub-figures are generated by training two RL agents based on two different values of the penalty factor ($\eta = 1$ and $\eta = 2$). After convergence (250 episodes), the average overall throughput and the average number of covered users are reported for five additional episodes in Figures 6a and 6b, respectively. We observe from these figures that increasing the penalty factor from 1 to 2 increases the average number of covered users by 0.24% and decreases the throughput by 5.2%, on average. Consequently, it is up to the service operator to adjust the penalty factor with the aim of striking an appropriate balance between the overall and individual experiences.

VI. Conclusions

In this work, we investigated the problem of self-optimizing the users' experience in a cellular network using reinforcement learning. To that end, we have recast the problem into an MDP. This entailed defining the state of the cellular network as a subset of relevant KPIs. We have introduced a novel action space, where both transmitted power and the relative CIOs of the eNBs are jointly controlled. Furthermore, we have introduced a new reward function, namely, the penalized throughput as a new measure of users' experience. The new metric reflects the tradeoff between the total throughput and the total number of covered users in the cellular network. Following this formulation, we propose using the TD3 reinforcement learning technique with carefully chosen hyper-parameters to learn the

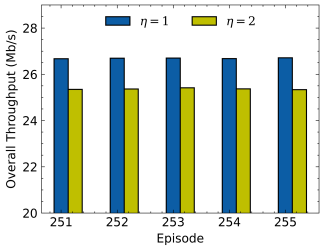


(a) Agent with penalized throughput reward

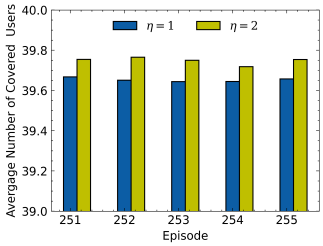


(b) Agent with throughput reward

Fig. 5: Radio environment map of the simulated environment controlled by two different agents



(a) Overall throughput



(b) Average Number of covered users

Fig. 6: Effect of changing penalty factor η

optimal power levels and CIOs from experience. Our technique has been tested in a simulated realistic setting using NS-3. The simulation setting admits 6 irregular eNBs functioning with the exact operator's parameters. Our numerical results showed impressive gains when using joint optimization of power levels and CIOs with respect to individual optimization of either of them and drastic gains relative to the baseline case. Furthermore, we introduced a method that allows a controllable tradeoff between the total throughput and the coverage of the cellular network.

REFERENCES

[1] Ericsson. Ericsson mobility report November 2019.

[2] K. M. Attiah, K. Banawan, A. Gaber, A. Elezabi, K. G. Seddik, Y. Gadallah, and K. Abdullah. Load balancing in cellular networks: A reinforcement learning approach. In *IEEE CCNC*, 2020.

[3] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang. What will 5G be? *IEEE JSAC*, 32(6):1065–1082, June 2014.

[4] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan. Load balancing in downlink LTE self-optimizing networks. In *IEEE VTC*, May 2010.

[5] S. S. Mwanje and A. Mitschele-Thiel. A Q-learning strategy for LTE mobility load balancing. In *PIMRC*, Sep. 2013.

[6] Y. Xu, W. Xu, Z. Wang, J. Lin, and S. Cui. Load balancing for ultradense networks: A deep reinforcement learning-based approach. *IEEE Internet of Things Journal*, 6(6):9399–9412, 2019.

[7] C. A. S. Franco and J. R. B. de Marca. Load balancing in self-organized heterogeneous LTE networks: A statistical learning approach. In *IEEE LATINCOM*, Nov 2015.

[8] P. Muñoz, R. Barco, J. M. Ruiz-Avilés, I. de la Bandera, and A. Aguilar. Fuzzy rule-based reinforcement learning for load balancing techniques in enterprise LTE femtocells. *IEEE Trans. on Vehicular Tech.*, 62(5):1962–1973, Jun 2013.

[9] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza. A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Comm. Surveys Tutorials*, 19(4):2392–2431, Fourthquarter 2017.

[10] S. Musleh, M. Ismail, and R. Nordin. Load balancing models based on reinforcement learning for self-optimized macro-femto LTE-advanced heterogeneous network. *Journal of Telecomm., Electronic and Computer Engineering (JTEC)*, 9(1):47–54, 2017.

[11] H. Zhang, X.-S. Qiu, L.-M. Meng, and X.-D. Zhang. Achieving distributed load balancing in self-organizing LTE radio access network with autonomic network management. In *IEEE Globecom Workshops*, 2010.

[12] A. Mukherjee, D. De, and P. Deb. Power consumption model of sector breathing based congestion control in mobile network. *Digital Communications and Networks*, 4(3):217–233, 2018.

[13] H. Zhou, Y. Ji, X. Wang, and S. Yamada. eicic configuration algorithm with service scalability in heterogeneous cellular networks. *IEEE/ACM Trans. on Networking*, 25(1):520–535, 2017.

[14] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1291–1307, November 2012.

[15] S. Fujimoto, H. Van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

[16] 3GPP ETSI TS 36.304 V15.5.0 . *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA), User Equipment (UE) procedures in idle mode*. 2019.

[17] 3GPP TR 32.836 V0.2.0. *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Study on NM Centralized Coverage and Capacity Optimization (CCO) SON Function (Release 12)*. 2012.

[18] 3GPP ETSI TS 136.213 V14.2.0. *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA), Physical layer procedures*. 2017.

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

[20] K. Abdullah, N. Korany, A. Khalafallah, A. Saeed, and A. Gaber. Characterizing the effects of rapid LTE deployment: A data-driven analysis. In *IEEE TMA*, 2019.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[22] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[23] A. Marella, A. Bonfanti, G. Bortolasor, and D. Herman. Implementing innovative traffic simulation models with aerial traffic survey. *Transport Infrastructure and Systems*, pages 571–577, 2017.

[24] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero. An open source product-oriented LTE network simulator based on ns-3. In *ACM MSWiM*, pages 293–298. ACM, 2011.

[25] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu. Stable baselines. https://github.com/hill-a/stable-baselines, 2018.

[26] P. Gawlowicz and A. Zubow. ns3-gym: Extending OpenAI Gym for Networking Research. *CoRR*, 2018.