Load Balancing and Energy Efficiency in Cellular Networks with a Scenario-Aware Reinforcement Learning Agent

Shorouk R. Abouamasha*, Mariam Aboelwafa†, Karim G. Seddik*
*Electronics and Communications Engineering, The American University in Cairo, Egypt
†Computer, Communications and Autonomous Systems Engineering, NewGiza University, Egypt
Email: shorouk_raafat@aucegypt.edu, mariam.nabil@ngu.edu.eg, kseddik@aucegypt.edu

***Abstract*—The vast proliferation of wireless data networks demands efficient resource allocation strategies to accommodate the increasing number of devices and the dynamic nature of cellular networks. As future networks face challenges like severe congestion and varying traffic demands, achieving satisfactory Quality of Service (QoS) and Quality of Experience (QoE) requires dynamic management. This paper introduces an improved self-optimization framework that adopts deep reinforcement learning (RL) to dynamically adjust key network parameters, such as handover settings, power levels, and MIMO technology. This approach significantly enhances network throughput by effectively balancing load distribution. The proposed framework explores the trade-off between system complexity and performance gains, demonstrating that an agent tailored to optimize a frequently recurring single scenario can outperform generalized agents under specific network conditions.**

***Index Terms*—Reinforcement Learning, Cellular Networks, Machine Learning, Load Balancing,**

I. Introduction

The surge in wireless data consumption from various devices necessitates the optimization of resource allocation in cellular networks. To support high-speed, low-latency applications and accommodate numerous connected devices [1], modern networks must manage resources effectively. Cellular networks aim for massive connectivity and data rates of up to 10 Gbps for low mobility and 1 Gbps for high mobility [2]. Future networks are expected to face severe congestion due to increased users, requiring high adaptability, self-organization, and rapid adjustment to maintain quality of service (QoS) and quality of experience (QoE). Achieving network stability while automatically optimizing parameters is complex and involves dynamic self-optimization based on traffic load. One approach is to modify cell boundaries and power levels to balance user distribution and enhance data throughput. This is achieved by altering cell individual offset (CIO) values [3], and/or dynamically adjusting base station power, which can effectively balance network load [4]. On another note, network operators prioritize optimizing power consumption. Since MIMO technology is a primary power consumer in cellular networks, studies [5], [6] have explored the potential energy savings from dynamically enabling and disabling MIMO based on network load.

In this paper, we examine the work introduced in [6] to modify the proposed approach and achieve better gains. Reinforcement Learning (RL) is adopted as a solution due to its ability to optimize long-term goals without requiring training data. In [6], a robust framework for the self-optimization of cellular networks was introduced using deep reinforcement learning. The goal is to improve network performance by balancing user load, enhancing coverage, improving user experience, and reducing energy consumption. The system introduced a Double Deep Q-Network (DDQN) agent followed by a Twin-Delayed Deep Deterministic (TD3) agent to adjust handover parameters, power levels, and MIMO technology. In this paper, we focus on redesigning the RL agents in [6] by introducing an additional continuous-action TD3 agent that is optimized for a *single, frequent scenario*. During the simulation, we observed that a certain case of the DDQN decision occurs with a higher probability than other cases. We considered this case in the training and optimized it in terms of power and CIO values using the TD3 agent. A notable improvement was observed when we used the scenario-aware agent when the state for which it was specifically trained for is encountered, revealing improved performance. This approach will result in a total of 3 agents, which will increase the overall system complexity, especially if multiple common scenarios exist, but it will be in favor of a significant increase in the overall network reward. To that end, it is noted that there is a trade-off between the number of agents used in the algorithm and the gain obtained. Optimizing network parameters with the scenario-aware agent outperforms the general agent optimizing all network scenarios in [6], making it an appealing option when designing the ML algorithm to self-optimize network parameters, especially when a scenario occurs more dominantly than others.

A. Related Work

As mentioned previously, this work aims to redesign the learning algorithm structure described in our previous work [6]. Building on it, we address three key network management controls: Cell Individual Offsets (CIOs), transmission power levels, and the activation of MIMO features. We compare the training of TD3 agents on multiple cases versus a single case of environment statuses, and how this affects the overall

network performance and results. The most relevant related works to this study focus on energy saving, load balancing issues, and different approaches for training RL agents.

The issue of energy-saving in cellular networks has been widely studied. For instance, in [7], the authors explored the dynamic operation of cellular base stations, proposing the deactivation of redundant stations during low-traffic periods, which leads to significant energy savings. In another study, [8] addressed energy optimization in mobile networks using a neural network-based algorithm, which activates the MIMO feature only when necessary to maintain satisfactory user quality of experience (QoE). Load-balancing techniques are also widely discussed in the literature. In [3] and [9], the authors developed an RL framework optimizing cell parameters to distribute traffic load evenly across cells. They focused on adjusting the CIOs of adjacent cells to encourage cell-edge users to switch from overloaded cells to those with lighter loads. Additionally, in [10] and [11], the authors presented an RL agent designed to control both the transmitted power of eNBs and the CIOs to aid in balancing traffic loads. The goal of the RL controller in these studies was to improve the downlink (DL) total throughput while minimizing the number of users who lack coverage.

In this work, we implement a layered RL agent using DDQN and TD3 algorithms [12], [13]. This layered approach is a natural extension of our previous DDQN and TD3 agents presented in [5], [6], [9]–[11]. However, our work focuses on evaluating the performance of the RL agent when trained on a single state of the environment, optimizing that state to take better actions compared to an agent trained on all states of the environment. A somewhat similar idea is discussed in [14], where the concept of role-oriented RL agents that can identify sub-tasks was introduced. This specialization facilitates performance improvement. The results show that gradually specialized roles are indispensable for performance enhancement. A HASSLE (Hierarchical Assignment of Subgoals to Subpolicies Learning) algorithm was introduced in [15] causing significant advancement in hierarchical reinforcement learning (HRL). HASSLE automatically discovers subgoals and develops specialized low-level policies, allowing high-level policies to set abstract subgoals based on observation clustering. The algorithm demonstrated superior performance compared to flat reinforcement learning methods in a simulated office navigation task, efficiently learning near-optimal policies. Additionally, this approach was proven to yield better performance in [16] in cases where useful subgoals can be identified and subtasks defined to achieve them.

B. Paper Contribution

The contribution of this paper can be summarized as follows:

- We present an improved RL-framework that controls handover parameters, transmission power levels, and MIMO scheme to maintain load balance within cellular networks to avoid both congestion and underutilization.
- We propose an improved hierarchical approach to make the control decisions, such that MIMO scheme control is decided first, affecting the decisions regarding the CIO and power level controls.
- We implement a scenario-aware RL agent approach that works along with the original agent in an alternating fashion based on the MIMO scheme decisions.

II. System Model

The main scenario is for a cellular system that serves a total of U User Equipment (UEs). The system comprises N base stations (eNBs).

A. Evolved Node B (eNodeBs "eNBs")

Each eNB emits a power level $P_n \in [P_{\min}, P_{\max}]$ dBm. At different times $t = 0, 1, 2, 3, \ldots$, each UE measures the Signal-to-Interference-plus-Noise-Ratio (SINR) of the near eNBs and attaches to the cell with the highest SINR. An eNB can be over-utilized or under-utilized. This is determined according to the value of the eNB utilization ρ_n:

$$\rho_n = \frac{\sum_{i=1}^{U_n} K_{i,n}}{B_n / B_{\text{PRB}}}, \tag{1}$$

where U_n is the number of UEs served by the nth eNB, $K_{i,n}$ is the number of Physical Resource Blocks (PRBs) that serve the ith user in the nth eNB, B_n is the bandwidth of the nth eNB and B_{PRB} is the bandwidth of one PRB (=180 KHz in LTE). ρ_n is the ratio of the total number of PRBs that the nth eNB must provide in order to serve the associated users to the maximum number of PRBs it can provide. Consequently, an underused eNB is roughly indicated by $\rho_n < 1$, whereas an overutilized eNB is indicated by $\rho_n > 1$. In contrast to overutilization, underutilization enables the eNB to provide reasonable rates for all of its associated customers.

Additionally, cells have an important power property which is the Cell Individual Offset (CIO) used to control the handover decision. Handover from cell i to a neighboring cell j occurs if [17]:

$$\text{RSRP}_j + \theta_{j-i} > \text{Hys} + \text{RSRP}_i + \theta_{i-j}, \tag{2}$$

where RSRP_i and RSRP_j are the measured Reference Signal Received Power from eNBs i and j, respectively. θ_{i-j} is the CIO value of eNB i with respect to eNB j and θ_{j-i} is the CIO value of eNB j with respect to eNB i. Hys is a hysteresis value to reduce the likelihood of recurrent handover requests brought on by small variations in signal quality. Moreover, network operators can enable or disable the MIMO feature for each eNB, significantly impacting the received rate. Enabling MIMO can reduce the BER through spatial diversity or increase the data rate via spatial multiplexing, improving UE QoE. However, MIMO is an energy-intensive feature.

B. User Equipment (UEs)

Each UE moves randomly in the network and is continuously seeking a better cell (based on the higher SINR), attaching itself to the superior cell when it finds it. Additionally, the connected cell receives periodic reports from the u^{th} UE's channel quality indicator (CQI) ϕ_u, which is a discrete metric. The standard states that $\phi_u \in \{0, 1, \ldots, 15\}$. If $\phi_u = 0$, the

u^{th} UE is not covered. Higher channel quality correlates with a higher CQI value [17], [18].

C. Control Agents Architecture and Decision Making

In this work, we improve upon the scheme presented in [6] by adopting two existing RL algorithms in a layered fashion. The first one is the Double Deep Q-Network (DDQN) [19], which is used for discrete action spaces (MIMO on/off in our case). The second one is Twin Delayed Deep Deterministic Policy Gradient (TD3) [20], which is used for continuous action spaces (transmission power and CIOs in our case). Our main focus in this work is improving the performance of the continuous action space agent.

We added a third scenario-aware TD3 agent which serves the same purpose as the first TD3 agent: controlling transmission power and CIOs. However, it is specifically designed to operate only when all eNBs are set to turn their MIMO feature on. This is one of the dominant decisions of the DDQN agent, particularly under highly congested scenarios, as seen later in Fig. 2 and Fig. 3

The *state* is defined as a subset of the network KPIs that are readily available to the network operator in practice. These KPIs include: Resource Block Utilization (RBU) $(B(t))$, total DL throughput of each cell $(R_n(t))$, number of active users in each cell $(C(t))$, Modulation and Coding Scheme (MCS) Matrix $(M(t))$. The MCS matrix represents the quality of the communication channels. The state is the concatenation of these vectors (after reshaping $M(t)$):

$$s(t) = [B(t)^T \quad R(t)^T \quad C(t)^T \quad \text{vec}(M(t))^T]. \tag{3}$$

where vec$(\cdot)$ represents matrix vectorization process.

The primary goal of the RL agent is to develop a policy that maximizes the expected reward over time [11]. This can be expressed as:

$$\max_{\pi} \lim_{L\to\infty} \sum_{t=0}^{L} E[\lambda^t r(t)], \tag{4}$$

where λ represents the discount factor that influences the weight given to future expected rewards. These formulations provide flexibility in choosing a reward function based on the operator's preferences. In this paper, we focus on the following **reward** function:

$$r(t) = \sum_{n=1}^{N} R(t) - \eta \bar{R}(t) \sum_{u=1}^{U} \mathbf{1}(\phi_u = 0) - \mu \sum_{n=1}^{N} m_n, \tag{5}$$

where η and μ are hyperparameters representing the penalties for user coverage and power consumption, respectively. They are selected to meet the operator's needs. $\bar{R}(t)$ represents the average user throughput at time t. This reward function is a linear combination of three terms. The first term $(\sum_{n=1}^{N} R(t))$ is the total network throughput (to be maximized). The second term $(\eta \bar{R}(t) \sum_{u=1}^{U} \mathbf{1}(\phi_u = 0))$ is the sum throughput of the uncovered users scaled by a hyper-parameter η. The final term $(\mu \sum_{n=1}^{N} m_n)$ represents a penalty as well, reflecting the number of eNBs with the MIMO feature activated. This penalty is controlled using a hyper-parameter μ. The selection of hyper-parameters depends on the operator's preferences. These preferences are influenced by the service provider's strategic goals (e.g., balancing coverage and capacity), channel conditions, and specific network configuration.

In this work, we show that having two RL agents and alternating between them: one specialized in a single case, trained to optimize the network for that specific scenario, and the second is the conventional agent trained for all other cases, results in better performance compared to the conventional agent alone. This approach is beneficial when a single scenario is observed to dominate the others.

III. Proposed Algorithm

An overview of the proposed scheme can be seen in Fig. 1. Decision-making occurs in two stages:

- **First Stage:** Based on the DDQN approach, the agent watches the state and decides when to turn MIMO on or off [19]. The discrete set $\{0, 1\}$ is used to execute the action (for each eNB).
- **Second Stage:** Based on the output of the first stage, a TD3 agent is selected to take the second stage decision. We augment the first-stage action with the observed state. The second stage decides the CIO and the variation in power levels based on the TD3 technique. If all the eNBs are turning MIMO on, then the dedicated agent for this case will be selected. The second TD3 agent will be chosen otherwise. TD3 actions are selected from the continuous intervals $[-\theta_{\min}, \theta_{\max}]$ and $[-P_{\min}, P_{\max}]$ respectively.

The proposed scheme is outlined in Algorithm 1, where $s(t)$ is the observed state at time t, $a_M(t)$ is the MIMO enabling action vector, $a_C(t)$ is the CIO values action vector and $a_P(t)$ is the transmitted powers action vector.

Algorithm 1 Proposed RL framework

```
Determine Reward Function.
Reset all values.
repeat
    procedure STAGE ONE
        Observe State (s(t)).
        Select MIMO feature decision (DDQN) (a_M(t)).
        Create a new augmented state (s_aug(t) = [s(t), a_M(t)]).
    end procedure
    procedure STAGE TWO
        Observe state (s_aug(t)) and select the proper TD3 agent according to a_M(t)
        Select relative CIO and power level actions from the chosen (TD3) ([a_C(t), a_P(t)])
        Apply augmented action to the network a_aug = [a_C(t), a_P(t), a_M(t)].
    end procedure
    Calculate Reward.
    Calculate the next state.
```

After the two stages, the augmented action is given by:

$$\begin{aligned} a(t) = &[(\theta_{ij} : i \neq j,\ i, j \in \{1, \cdots, N\}, \\ &(P_n : n \in \{1, 2, \cdots, N\}), \\ &(m_n : n \in \{1, 2, \cdots, N\})] \end{aligned} \tag{6}$$

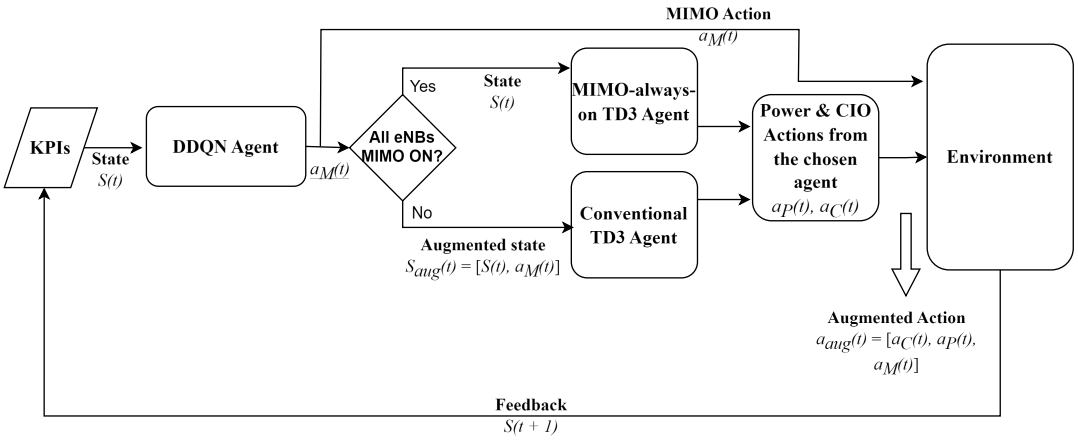

Fig. 1. An overview of the proposed algorithm decision-making process.

$a(t)$ is then applied to the environment.

As the agent explores the whole action space, the scenario-aware TD3 agent learns to optimize the transmitted power and CIO values of the eNBs when all eNBs are operating their MIMO feature (when the first agent decides to turn MIMO on for all eNBs). The second, traditional TD3 agent learns to optimize power and CIO values for other combinations of the first-stage actions (MIMO on/off). The specific scenario agent is activated only when the scenario, for which it was designed and trained, occurs; otherwise, the general conventional TD3 agent is used.

IV. Performance Evaluation

A. Network Simulator

The proposed approach involves using RL agents to optimize cellular networks without prior knowledge of the optimal policy, requiring them to learn through interaction. The approach utilizes the NS3 network simulator and specifically its LTE module to accurately emulate the LTE system. Modifications to NS3 allow the agent to control cell CIOs and choose MIMO modes. The NS3gym interface connects NS3 to the OpenAI Gym, facilitating RL for network optimization by handling the exchange of actions, states, and rewards between the agents and the environment. The implementation of the RL agents (specifically, the TD3 agents) is done using Python implementations of the stable-baselines3 library.

B. Simulation Setup:

For the simulation[1], we chose a $900\text{m}\times1800\text{m}$ area in the urban Fifth Settlement neighborhood in Egypt. Within this area, we used a realistic placement of eNodeBs to form a network cluster of six eNBs, with locations specified by one of the 4G network operators in Egypt. We have 6 eNBs and the DDQN agent controls the MIMO feature operation of each of them. Similar parameters to those used in [6] are used. The Key Performance Indicator (KPI) we used to monitor progress and enhancements in our study is the total downlink throughput per cell. To simulate realistic user mobility in our environment, we utilize the Simulation of Urban Mobility (SUMO) tool [21], which is known for its simplicity and effectiveness in generating realistic movement patterns. SUMO can import precisely emulated environments from real-world maps. We use the SUMO simulator to implement realistic mobility models for the UEs, which include both vehicles and pedestrians. Pedestrians move at speeds between $0-3$ m/s, while vehicle movement parameters such as acceleration, deceleration, speed factor, and speed deviation are based on data from [22] to mimic realistic vehicle behavior. UEs are initially placed randomly on available streets and pedestrian pathways. During the simulation, each UE takes a random trip from a starting point to a destination. Users are assumed to follow a full-buffer traffic model, meaning they are always active.

We extend the simulated cellular network presented in [5] and [6]. In addition to the previous work, we introduce another TD3 agent that is trained on an environment where all eNodeBs have their MIMO feature always turned on. This is because some MIMO decision combinations occur more frequently than others, especially with increased congestion and mobility in the network. In the case of having a penalty on user coverage $\eta = 2$ value, a histogram of the number of eNBs having MIMO-on feature is shown in Fig. 2. The x-axis represents the total number of eNBs (out of 6) turning MIMO on during the simulation time. In the second case, shown in Fig. 3, there is no penalty in the reward function ($\mu = 0$ and $\eta = 0$), i.e., the reward function targets maximizing the sum throughput. In this case, most of the eNBs have MIMO-on with high frequency, as maximizing the sum throughput favors switching MIMO on. For this reason, we choose to specialize a scenario-aware agent to optimize the transmitted power and CIO values for the eNBs when all eNBs are operating in MIMO mode.

Based on the previous results, testing the scenario-aware agent shows a significant increase in the downlink throughput compared to the TD3 agent trained for all DDQN agent

[1]The Codes for this work are readily available at the following Github repository: https://github.com/shoroukraafat/Load-balancing-in-cellular-networks-with-a-scenario-aware-RL-agent

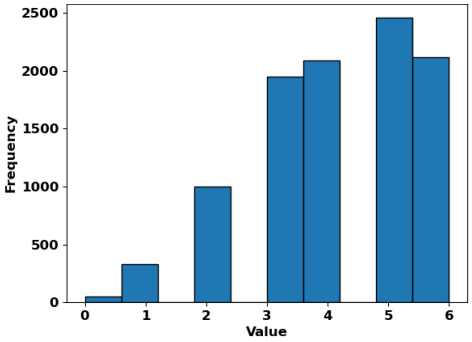

Fig. 2. The frequency of MIMO-on state for all eNBs for penalized throughput case for $\eta = 2$ (number of runs = $10k$)

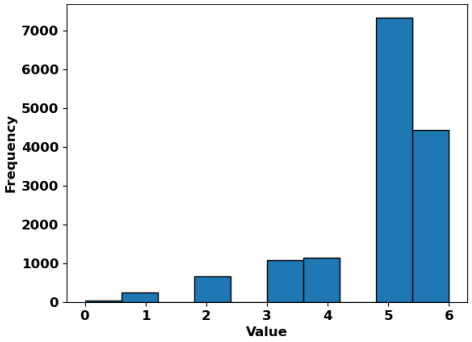

Fig. 3. The frequency of MIMO-on state for all eNBs with no penalty for uncovered users case for $\eta = 0$ (number of runs = $15k$)

decisions. This occurs when the environment operates with the MIMO feature turned on for all base stations, which is a case that occurs more frequently.

C. Results

In this section, we assess the performance of our proposed approach by testing the effect of different hyperparameters on the sum throughput of the network. Both TD3 agents (the conventional TD3 agent from [6] and the scenario-aware TD3 agent) are trained to get the best model. Testing scenarios are as follows:

1) *MIMO always on with penalty on the user coverage:* We employ a RL agent to optimize the CIOs and power levels only. We switch on the MIMO feature at all times for all eNBs and use a penalty on uncovered users with $\eta = 2$ in equation (5). This setup allows us to compare the performance of both agents in the same environment when all eNBs are forced to keep MIMO on to observe the agents' performance in this specific case. We plot the network sum throughput (in Mbps) versus steps in Fig. 4 for a testing episode of 250 steps. Our proposed algorithm achieves a gain of approximately 4 Mbps over the conventional agent.
2) *MIMO always on with no penalty on the user coverage:* This scenario is similar to the previous one but we set no penalty on uncovered users with $\eta = 0$. This is done to compare the performance of both agents (scenario-aware and conventional) based solely on pure throughput values, without the influence of other factors. The gain of our proposed agent is higher in this case as shown in Fig. 5 compared to the case simulated in Fig. 4; the throughput gain is mostly above $7Mbps$.
Since better throughput results were obtained under the case of no coverage penalty, we used the scenario and models trained in the absence of this penalty in the reward function, i.e., we set $\eta = 0$, to simulate the performance of our proposed agent against the conventional agent in the environment where MIMO switching decisions are made by the DDQN agent.

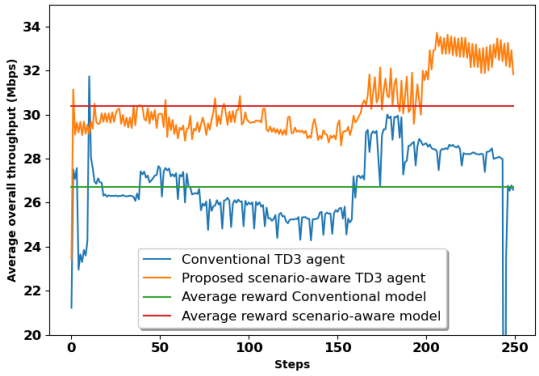

Fig. 4. Comparison of the sum throughput reward for the MIMO-on agent and conventional agent for $\eta = 2$

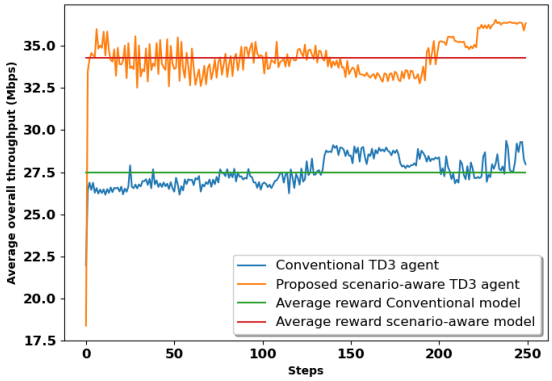

Fig. 5. Comparison of the sum throughput reward for the MIMO-on agent and conventional agent for $\eta = 0$

3) *MIMO on/off with no penalty on the user coverage:* In this simulation scenario, the DDQN agent first decides whether to enable the MIMO features of eNBs. If MIMO is enabled for all eNBs, the scenario-aware TD3 agent (trained specifically for this all-MIMO-on scenario) decides the continuous actions for CIOs and transmitted powers. Otherwise, the conventional TD3 agent from [6] is used to decide on the CIOs and power values. The results of this case are shown in Fig. 6.

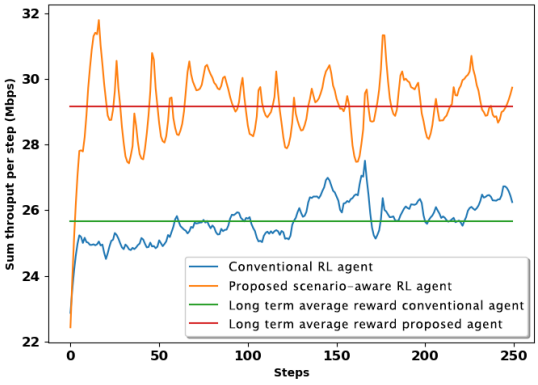

Fig. 6. Average throughput per step over 20 episodes while alternating between conventional and MIMO-on agents for $\eta = 0$

Fig. 6 shows the average sum throughput per step, averaged over 20 episodes, and it reveals that the proposed algorithm, which introduces a MIMO-ON-only agent that alternates with the conventional agent, achieves significantly better performance than the conventional TD3 agent in the simulated scenario. The long-term average difference between our proposed algorithm and the conventional one is approximately $3Mbps$. This increase in sum throughput comes with system complexity, presenting a performance-complexity trade-off for system designers when selecting the network RL agents.

V. Conclusion

In this paper, we introduced a scenario-aware TD3 agent that is trained to make optimal decisions in the case of all eNBs enabling the MIMO feature. This case is selected because of its significant occurrence in the environment during the simulation time. The added agent works along with the conventional agent to provide better performance when each is used in the case for which it was trained. This suggestion introduces some intricacy to the design but favors an increase in total throughput. This trade-off should be assessed by system designers according to the environment conditions to adopt the new approach when a case occurs much more frequently than others. Specializing an agent for this case will result in better results for the agent's actions. While the paper demonstrates the benefits of scenario-specific RL agents, future work should focus on exploring their complexity-performance trade-off and scalability across diverse and dynamic cellular network conditions.

References

[1] P. Varga, J. Peto, A. Franko, D. Balla, D. Haja, F. Janky, G. Soos, D. Ficzere, M. Maliosz, and L. Toka, "5g support for industrial iot applications— challenges, solutions, and research gaps," *Sensors*, vol. 20, no. 3, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/3/828

[2] A. Ercan, M. O. Sunay, and I. F. Akyildiz, "Rf energy harvesting and transfer for spectrum sharing cellular iot communications in 5g systems," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1680–1694, 2018.

[3] K. Attiah, K. Banawan, A. Gaber, A. Elezabi, K. Seddik, Y. Gadallah, and K. Abdullah, "Load balancing in cellular networks: A reinforcement learning approach," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–6.

[4] 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 Telecommunication, Electronic and Computer Engineering*, vol. 9, pp. 47–54, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:52242424

[5] M. Aboelwafa, G. Alsuhli, K. Banawan, and K. G. Seddik, "Self-optimization of cellular networks using deep reinforcement learning with hybrid action space," in *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, 2022, pp. 223–229.

[6] B. Salama Attia, A. Elgharably, M. Nabil Aboelwafa, G. Alsuhli, K. Banawan, and K. G. Seddik, "Self-optimized agent for load balancing and energy efficiency: A reinforcement learning framework with hybrid action space," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 4902–4919, 2024.

[7] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 56–61, 2011.

[8] M. Aboelwafa, M. Zaki, A. Gaber, K. Seddik, Y. Gadallah, and A. Elezabi, "Machine learning-based mimo enabling techniques for energy optimization in cellular networks," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–6.

[9] G. Alsuhli, K. Banawan, K. Attiah, A. Elezabi, K. G. Seddik, A. Gaber, M. Zaki, and Y. Gadallah, "Mobility load management in cellular networks: A deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 22, no. 3, pp. 1581–1598, 2023.

[10] G. Alsuhli, H. A. Ismail, K. Alansary, M. Rumman, M. Mohamed, and K. G. Seddik, "Deep reinforcement learning-based cio and energy control for lte mobility load balancing," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021, pp. 1–6.

[11] G. Alsuhli, K. Banawan, K. Seddik, and A. Elezabi, "Optimized power and cell individual offset for cellular load balancing via reinforcement learning," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–7.

[12] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.06461

[13] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018. [Online]. Available: https://arxiv.org/abs/1802.09477

[14] T. Wang, H. Dong, V. Lesser, and C. Zhang, "Roma: Multi-agent reinforcement learning with emergent roles," 2020. [Online]. Available: https://arxiv.org/abs/2003.08039

[15] B. Bakker and J. Schmidhuber, "Hierarchical reinforcement learning with subpolicies specializing for learned subgoals," 01 2004, pp. 125–130.

[16] T. G. Dietterich, "Hierarchical reinforcement learning with the maxq value function decomposition," 1999. [Online]. Available: https://arxiv.org/abs/cs/9905014

[17] E. LTE, "Evolved universal terrestrial radio access (e-utra); physical layer procedures (3gpp ts 36.213 version 14.6.0 release 14)," *ETSI TS*, pp. 136–213, 2018.

[18] E. TSGR, "Lte: Evolved universal terrestrial radio access (e-utra)," *Multiplexing and channel coding (3GPP TS 36.212 version 10.3. 0 Release 10) ETSI TS*, vol. 136, no. 212, p. V10, 2011.

[19] Q. Zhang, T. Du, and C. Tian, "A sim2real method based on ddqn for training a self-driving scale car," *Mathematical Foundations of Computing*, vol. 2, pp. 315–331, 01 2019.

[20] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour *et al.*, "Policy gradient methods for reinforcement learning with function approximation." in *NIPs*, vol. 99. Citeseer, 1999, pp. 1057–1063.

[21] D. Krajzewicz and C. Rossel, "Simulation of urban mobility (sumo)," *Centre for Applied Informatics (ZAIK) and the Institute of Transport Research at the German Aerospace Centre*, 2007.

[22] A. Marella, A. Bonfanti, G. Bortolasor, and D. Herman, "Implementing innovative traffic simulation models with aerial traffic survey," *Transport infrastructure and systems*, pp. 571–577, 2017.