Optimisation of Resource Allocation in Heterogeneous Wireless Networks Using Deep Reinforcement Learning

Oluwaseyi Giwa
Mathematical Sciences
African Institute for Mathematical Sciences
South Africa
oluwaseyi@aims.ac.za

Jonathan Shock
Department of Mathematics and Applied Mathematics
University of Cape Town, and
NiTheCS, Stellenbosch
South Africa
INRS, Montreal
Canada
jonathan.shock@uct.ac.za

Jaco du Toit
AI/ML & Data Technology Strategy and Assurance
Vodacom, and
EEE, Stellenbosch
South Africa
jacowp357@gmail.com

Tobi Awodumila
AI4Science
African Institute for Mathematical Sciences
South Africa
tobi@aims.ac.za

***Abstract*—Dynamic resource allocation in heterogeneous wireless networks (HetNets) is challenging for traditional methods under varying user loads and channel conditions. We propose a deep reinforcement learning (DRL) framework that jointly optimises transmit power, bandwidth, and scheduling via a multi-objective reward balancing throughput, energy efficiency, and fairness. Using real base station coordinates, we compare Proximal Policy Optimisation (PPO) and Twin Delayed Deep Deterministic Policy Gradient (TD3) against three heuristic algorithms in multiple network scenarios. Our results show that DRL frameworks outperform heuristic algorithms in optimising resource allocation in dynamic networks. These findings highlight key trade-offs in DRL design for future HetNets.**

***Index Terms*—Resource Allocation, Deep Reinforcement Learning, Heterogeneous Networks.**

I. Introduction

The evolution towards fifth-generation (5G) and the forthcoming sixth-generation (6G) wireless systems is driven by a demand for ubiquitous connectivity and high data rates. This has led to the proliferation of Heterogeneous Networks (HetNets), which overlay traditional macrocells with dense tiers of small cells (e.g., micro, pico, and femto cells) to enhance spectral efficiency and network capacity [1], [2]. However, this architectural complexity introduces challenges in resource allocation (RA). The dense deployment of base stations (BS) creates severe co-tier and cross-tier interference, making the efficient management of spectrum, transmit power, and user association critical for network performance. Optimising these resources is essential not only to maximise throughput but also to ensure fairness and quality of service (QoS) for all users in the network [3], [4].

Traditional RA strategies, relying on classical optimisation or heuristics [5], are inadequate for modern HetNets [6]. These methods depend on simplified, static network models and struggle with the nonconvex, combinatorial nature of joint RA problems. The dynamic reality of wireless networks, characterised by fluctuating user mobility, channel conditions, and traffic loads [7], renders static assumptions invalid and leads to suboptimal solutions, necessitating a shift to intelligent, data-driven frameworks [8].

Reinforcement learning (RL) has emerged as a powerful paradigm for this challenge. By learning policies through direct environmental interaction [9], RL agents adapt to real-time conditions without an explicit model. Recent deep reinforcement learning (DRL) approaches effectively handle the high-dimensional state and action spaces of modern networks [10]–[18], demonstrating superior performance over rule-based methods in tasks ranging from power control to network slicing.

In this paper, we propose a DRL framework to jointly optimise transmission power, bandwidth, and user scheduling in a dynamic HetNet. We move beyond existing works by providing a direct comparative analysis of two state-of-the-

This work was supported by the African Institute for Mathematical Sciences (AIMS), South Africa, and the Mastercard Foundation Scholarship.
The work of Tobi Awodumila has received funding from the Google DeepMind Scholarship under AIMS.

art DRL algorithms: Twin Delayed Deep Deterministic Policy Gradient (TD3) and Proximal Policy Optimisation (PPO). DRL is well-suited for this problem's continuous action spaces and complex policy learning. TD3 is chosen for its sample efficiency and stability, while PPO is selected for its robustness and ease of implementation. Our comparison uncovers critical trade-offs between performance, stability, and convergence speed.

Our key contributions are threefold. First, we formulate the multi-objective RA problem as a Markov Decision Process (MDP) and develop a custom, publicly available simulation environment for reproducibility[1]. Second, we implement and evaluate TD3 and PPO, analysing their effectiveness in balancing throughput, fairness, and energy consumption. Third, our experiments provide practical insights into the strengths of each algorithm, offering guidance for selecting algorithms based on network priorities. Finally, we compare our DRL agents against three heuristic algorithms. The remainder of this paper is organised as follows: Section II details the system model and problem formulation. Section III describes the DRL algorithms. Section IV presents the experimental setup. Section V discusses the results, and Section VI concludes the paper.

II. System Model

A. Assumptions and MDP Formulation

We study *downlink* RA[2] in a heterogeneous cellular network with N_M macro BS and N_S micro BS for a total $N_B = N_M + N_S$ stations, serving N_U users distributed within the coverage area. We model control as an MDP and the RA controller as a single agent that interacts with the HetNet environment (Fig. 1) at discrete time steps $t, \quad \text{for } t \in [0, 1, \ldots, T-1]$. This is done because the network's state at each time step fully captures the necessary information to model the system's evolution based on the agent's action, satisfying the Markov property.

$$\mathcal{M} = \big(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma\big), \tag{1}$$

where $s_t \in \mathcal{S}$ is the state, $a_t \in \mathcal{A}$ is the action, $\mathcal{P}$ is the transition kernel, R is the reward, and $\gamma \in (0, 1)$ is the discount factor.

State Space ($\mathcal{S}$): The state, $s_t \in \mathcal{S}$, is a high-dimensional vector that concatenates all network observables and summaries needed for control.

$$s_t = \Big[\vec{p_t}, \vec{I_t}, \mathbf{A}_t, \vec{x_t}^{\text{BS}}, \vec{x_t}^{\text{U}}\Big]_{t \in T, \text{BS} \in N_B, \text{U} \in N_U}, \tag{2}$$

Where $\vec{p_t}$ is a vector of the current power levels for all N_B BSs. $\vec{I_t}$ is a vector containing the estimated interference experienced by each of the N_U users. $\mathbf{A}_t$ is the matrix representing the current BS-user associations and allocated resources. $\vec{x_t}^{\text{BS}}$ and $\vec{x_t}^{\text{U}}$ represent the fixed locations of the BSs and user equipment(UE), respectively.

Action Space ($\mathcal{A}$): The action, $a_t \in \mathcal{A}$ is a composite vector of control decisions for all N_B BSs. For each BS $\in N_B$, the decision variables are the transmission power adjustment ($p_{\text{BS}}^{\text{adj}}$), the fraction of total bandwidth to allocate ($w_{\text{BS}}^{\text{alloc}}$), and a scheduling score ($s_{\text{U}}^{\text{score}}$) indicating user priority.

$$a_t = \left\{ \left(p_{\text{BS}}^{\text{adj}}, w_{\text{BS}}^{\text{alloc}}, s_{\text{U}}^{\text{score}} \right) \right\}_{\text{BS}=1}^{N_B} \tag{3}$$

To facilitate a stable learning process, these action components are normalised to the range. This bounding of the agent's output is critical for neural network-based function approximators, as it prevents exploding gradients and allows the agent to learn a generalised policy independent of the physical units. The environment then translates these normalised values into their corresponding physical domains via affine transformations. For instance, the actual transmission power for a particular BS is mapped from the normalised agent output, $a_p \in$ as:

$$P_{\text{BS}} = P_{\text{min}} + a_p \cdot (P_{\text{max}} - P_{\text{min}}) \tag{4}$$

Where P_{min} and P_{max} are the minimum and maximum power levels, respectively. A similar linear mapping is applied to the bandwidth allocation.

Channel Dynamics ($\mathcal{P}$): Our channel dynamics are modeled using stochastic wireless parameters like Signal-to-Interference-plus-Noise Ratio (SINR), path loss exponent (η), log-normal shadowing ($\Psi_{\text{BS-U}}$), downlink effective power gain ($H_{\text{BS-U}}$), and noise power (N_0). In our SINR model, the signal component depends on the serving BS's transmit power, distance, and shadowing, while the interference term aggregates power from all non-serving BSs. This formulation accurately reflects the user's experienced link[3] quality, driving reward calculation and throughput estimation.

$$\Psi_{\text{BS-U}} = 10^{X_{\text{BS-U}}/10}, \quad X_{\text{BS-U}} \sim \mathcal{N}\left(0, \sigma_{\text{sh}}^2\right) \text{ in dB} \tag{5}$$

$$H_{\text{BS-U}} = \frac{S_{\text{BS-U}}}{(d_{\text{BS-U}})^{\eta} \cdot \Psi_{\text{BS-U}}} \tag{6}$$

Where $X_{\text{BS-U}}$ is a zero-mean Gaussian in decibels (dB) with standard deviation σ_{sh}^2. A positive X increases path loss (deep fade) and a negative one decreases it (shadowing "gain"). $S_{\text{BS-U}}$ and $d_{\text{BS-U}}$ are the fading factor (dimensionless) and distance between BS and user, respectively.

$$\text{SINR}_{\text{U}} = \frac{p_{\text{BS-U}} \cdot H_{\text{BS-U}}}{\sum_{\text{BS'} \neq \text{BS}} (p_{\text{BS-U}} \cdot H_{\text{BS-U}}) + N_0} \tag{7}$$

The per-user throughput is calculated using the Shannon capacity formula [19], scaled by the allocated bandwidth B_{U}:

$$T_{\text{U}} = B_{\text{U}} \cdot \log_2 (1 + \text{SINR}_{\text{U}}) \tag{8}$$

B_{U} is determined by the agent's bandwidth allocation action $w_{\text{BS}}^{\text{alloc}}$ and scheduling score $s_{\text{BS}}^{\text{score}}$.

Unit Consistency: All power related quantities (e.g., $p_{\text{BS-U}}, N_0$ are handled in linear scale (mW). Distances are computed in meters.

Reward function ($\mathcal{R}$): The agent's learning is guided by a multi-objective reward function $r_t = \mathcal{R}(s_t, a_t)$, designed

[1] GitHub repo

[2] Note that in an uplink communication scenario, the users transmit to the BS. This would require modifications to our MDP formulation.

[3] A link in this context is the interaction between a BS and a user

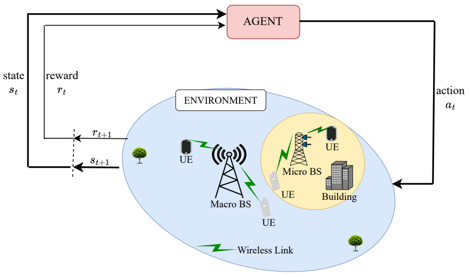


Fig. 1: An illustration of the RL agent in the HetNet environment.

to balance three competing network objectives: maximising aggregate throughput, ensuring fairness, and minimising power consumption. The reward at time step t is a weighted linear combination:

$$r_t = \kappa \cdot \sum_{\mathrm{u}=1}^{N_U} T_{\mathrm{U}} \;-\; \beta \cdot \sum_{\mathrm{BS}=1}^{N_B} P_{\mathrm{BS}} \;+\; \phi \cdot \text{Fairness}_t \tag{9}$$

The weighting coefficients κ, β, ϕ are hyperparameters tuned to reflect the desired network operating priorities. The Fairness is calculated using Jain's fairness index [20]:

$$\text{Fairness}_t \triangleq J = \frac{\left(\sum_{\mathrm{U}=1}^{N_U} z_{\mathrm{U}}\right)^2}{N_U \sum_{\mathrm{U}=1}^{N_U} z_{\mathrm{U}}^2} \tag{10}$$

Where z_{U} is the amount of resource allocated to a user.

Optimisation objective: The goal of the DRL agent is to learn an optimal policy $\pi*$ that maps states to actions, maximising the expected cumulative discounted reward:

$$\pi* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{L} \gamma^t r_t | \pi\right] \tag{11}$$

Where L is the episode horizon and $\gamma \in [0, 1)$ is the discount factor, which balances the trade-off between immediate and future rewards.

B. Satellite-Derived Topology

We instantiate BS locations from real BS location data in Cape Town, provided by a local telecom operator and place 50 users within the deployment polygon. The dataset includes three macro BSs and ten micro BSs. Fig. 2 shows the satellite view used to derive the layout. Colors in all figures follow the evaluation convention: Macro BS (red), Micro BS (blue), Users (yellow).

III. DRL Algorithms

The RA problem formulated in Section II is characterised by a high-dimensional state space and a continuous action space (for transmit power and bandwidth). This renders DRL algorithms, such as Deep Q-Networks (DQN), which are restricted to discrete actions, unsuitable. Consequently, we turn to actor-critic and policy-gradient methods, which are designed for continuous control. While Deep Deterministic Policy Gradient (DDPG) is a natural starting point, it is known to suffer from instability and Q-value overestimation. We therefore select two state-of-the-art algorithms that address these challenges: TD3, which directly mitigates the shortcomings of DDPG, and PPO, which is renowned for its robustness and stable training performance.

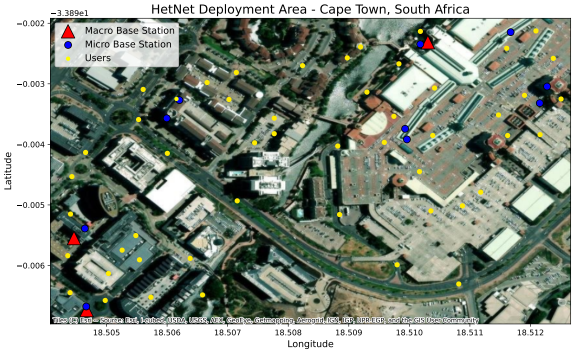

HetNet Deployment Area - Cape Town, South Africa
−3.389e1
−0.002
−0.003
−0.004
−0.005
−0.006
Latitude
18.505
18.506
18.507
18.508
18.509
18.510
18.511
18.512
Longitude
Macro Base Station
Micro Base Station
Users
Tiles (C) Esri -- Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community

Fig. 2: Satellite image of the deployment area. Source: Esri GIS software for mapping and spatial analysis.

A. Twin Delayed Deep Deterministic Policy Gradient (TD3)

TD3 is an off-policy, model-free algorithm that builds upon DDPG by introducing several key modifications to enhance stability and performance. It learns a deterministic policy (the actor) that maps states to actions, and a Q-function (the critic) that estimates the action-value function. The three core innovations of TD3 are:

Clipped Double Q-Learning: To combat the overestimation bias of the critic, TD3 employs two independent critic networks, Q_{θ_1} and Q_{θ_2}. When computing the target value for the Bellman update, it takes the minimum of the two critics' predictions, yielding a more conservative and stable target:

$$y = r + \gamma \min_{i=1,2} Q_{\theta'_i}\left(s', \pi_{\mu'}(s') + \epsilon\right) \tag{12}$$

Where μ' and θ' are the parameters of the target networks, and the noise ϵ is for target policy smoothing.

Delayed Policy Updates: The actor network (π_μ) is updated less frequently than the critic networks. This allows the critic's Q-value estimates to converge and stabilise before being used to update the actor, leading to more reliable policy improvements.

Target Policy Smoothing: Noise is added to the target action during the target Q-value calculation. This helps to regularise the policy, making it less likely to exploit narrow peaks in the value function, which results in a smoother policy landscape.

B. Proximal Policy Optimisation (PPO)

PPO is an on-policy, actor-critic algorithm celebrated for its balance of sample efficiency and ease of implementation. Unlike TD3, PPO learns a stochastic policy, $\pi_\theta(a|s)$. Its key feature is a novel surrogate objective function that constrains the size of policy updates, preventing destructive, large changes during training. The core of PPO is the clipped surrogate objective function, which modifies the standard policy gradient objective. It uses the ratio between the new policy and the old policy, $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, to measure the policy change. The objective is:

$$L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \tag{13}$$

Where $\hat{A}_t$ is an estimate of the advantage function (often computed using Generalised Advantage Estimation, GAE), and ϵ is a small hyperparameter that defines the clipping range. This objective clips the probability ratio, which discourages policy updates that move $r_t(\theta)$ far from 1, thereby ensuring more stable training. See Appendix for TD3 and PPO training loop.

IV. Experimental Scenarios

We evaluated our DRL agent in multiple scenarios. While user positions are static within an episode, they are re-randomised for each new episode to ensure the agents learn generalisable policies. Both algorithms were trained over 10 training seeds ($\{0, 10, 18, 28, 42, 64, 128, 256, 512, 1024\}$) for 1000 episodes, with each episode comprising 1000 discrete timesteps. Refer to Table II in the Appendix, which summarises the values of the parameters and hyperparameters used for our experiments. While some parameters were fixed (e.g., path loss exponent η), others, such as the weighting coefficients (κ, β, ϕ), learning rate (α), and GAE parameter λ, were fine-tuned by experimenting with different set of values to obtain the best result.

A. Evaluation Metrics

To evaluate the learning and performance of the agents, we use the following key metrics, which directly correspond to the objectives of the resource allocation problem:

Average Reward: This is the primary metric for assessing overall performance. It represents the cumulative, discounted reward obtained by the agent over an episode. A higher and more stable reward indicates that the agent has successfully learned a policy that balances the multi-objective function of maximising throughput, ensuring fairness, and maintaining energy efficiency.

Transmit Power Allocation: This metric reflects the agent's learned policy for energy efficiency. It is the normalised transmit power level (between 0 and 1) that the agent allocates per base station.

Bandwidth Allocation: This metric measures the agent's strategy for spectral resource management. It is the normalised bandwidth fraction allocated per base station.

Scheduling Score: This metric serves as a proxy for fairness. It is the normalised score assigned per user, which influences their priority for resource allocation.

V. Performance Comparison and Discussion

The agents' learning is compared in the mean reward convergence plot (Fig. 3d). The TD3 agent learns faster initially; its sample efficiency, from a prioritised experience replay buffer, makes it suitable for rapid deployment. In contrast, PPO shows a slower but more stable and superior learning trajectory. Its conservative and stochastic policy updates encourage exploration, making PPO preferable for achieving long-term, near-optimal performance.

Figs. 3a to 3c further support this, which illustrate various wireless network parameters, including bandwidth and transmit power allocation, as well as user fairness.

A. Comparison Against Heuristic Baselines

To ground the DRL results, we compare them against three heuristics prioritising a single goal under the same constraints and observables. The baselines are: Greedy OFDMA-like (G-OFDMA) for throughput, Interference-Pricing Power Control (IP-PC) for power reduction, and Proportional-Fair with Equal Bandwidth (PF-EQ) for fairness. Table I compares them across four scenarios.

G-OFDMA (throughput-first): Each BS greedily schedules the user with the best link, allocating most bandwidth and operating near its power budget to maximise sum-rate. This simple strategy boosts instantaneous throughput but neglects weaker users and increases inter-cell interference.

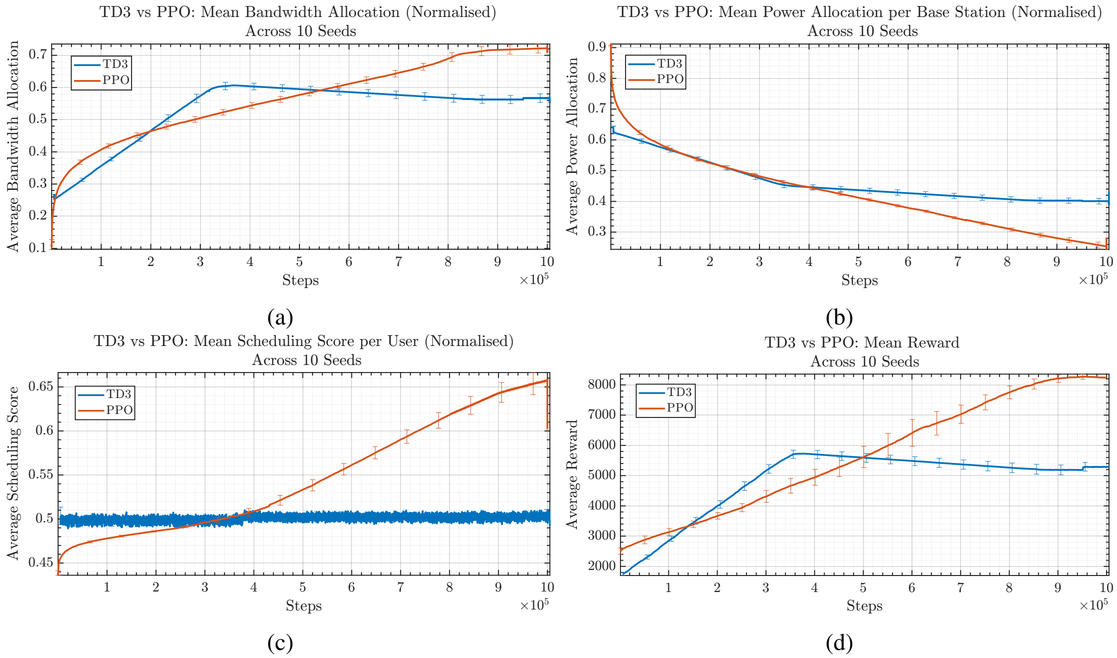
IP-PC (power-first): This baseline reduces transmit power where it causes high interference to neighbours. Using measured interference as a "price," each BS lowers its power where the marginal cost is high, using simple (strongest-BS) association and conservative bandwidth allocation. This yields low power and interference leakage but sacrifices throughput under load.

PF-EQ (fairness-first): Each BS splits its bandwidth evenly among its attached users, employing a proportional-fair scheduler to prioritise users with recently poor service. Power is maintained at a moderate level. This approach improves user fairness and queue stability but underutilises peak capacity compared to opportunistic schedulers.

Discussion across scenarios: Across all scenarios, the heuristics reveal expected trade-offs: G-OFDMA favours throughput at the cost of fairness and power, IP-PC minimises power, and PF-EQ improves fairness but underuses capacity. In contrast, the DRL agents learn to jointly adapt power, bandwidth, and user association over time, finding operating points that more effectively balance reward, fairness, and energy. This consistent superiority highlights that the DRL gains are not tied to a single topology.

VI. Conclusion

In this paper, we addressed the resource allocation problem in heterogeneous wireless networks by presenting a comparative analysis of the PPO and TD3 deep reinforcement learning algorithms. Our findings, based on a realistic network deployment, reveal that while TD3 demonstrates faster initial convergence, PPO achieves a significantly higher overall reward



(a)

(b)

(c)

(d)

Fig. 3: Comparison of the mean performance of PPO and TD3. Error bars represent 95% confidence interval computed across 10 training seeds over 1 million steps. **(a)** Normalised average bandwidth allocation. **(b)** Normalised average transmit power allocation. **(c)** Average scheduling score demonstrating. **(d)** Average reward convergence.

by learning more effective policies for energy conservation and user fairness. This highlights a critical trade-off: TD3 is a sample-efficient algorithm suitable for rapid deployment, whereas PPO's methodical exploration yields a more globally optimal policy for performance-critical applications. Future work will focus on extending this framework to multi-agent scenarios and incorporating the effects of user mobility.

ACKNOWLEDGEMENT

We thank Claude Formanek for his initial assistance with the algorithm design.

APPENDIX

We provide the algorithms for both TD3 (Algorithm 1) and PPO (Algorithm 2) for the resource allocation problem. In addition, Table II provides a list of parameters used in the experiments.

Algorithm 1 TD3 for Resource Allocation Optimisation

```
Initialize actor $\pi_\mu$, critics $Q_{\theta_1}$, $Q_{\theta_2}$, and their target networks $\pi_{\mu'}, Q_{\theta'_1}, Q_{\theta'_2}$ and replay buffer $\mathcal{D}$.
for each training step do
    Select action with exploration noise: $a = \pi_\mu(s) + \mathcal{N}(0, \sigma)$.
    Store $(s, a, r, s')$ in $\mathcal{D}$ and sample a minibatch from $\mathcal{D}$
    Compute target action with smoothed noise: $a' \leftarrow \pi_{\mu'}(s') + \text{clip}\,(\mathcal{N}(\prime, \sigma), -c, c)$.
    Compute target Q-value: $y = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', a')$
    Update critics $\theta_i$ by minimizing Huber/MSE loss: $\mathcal{L}(\theta_i) = (Q_{\theta_i}(s, a) - y)^2$.
    if step is a policy update step then
        Softly update all target networks: $\theta' \leftarrow \tau\theta + (1 - \tau)\theta', \mu' \leftarrow \tau\mu + (1 - \tau)\mu'$.
    end if
end for
```

REFERENCES

[1] X. Yongjun, G. Guan, G. Haris, and A. Fumiyuki, "A survey on resource allocation for 5G heterogeneous networks: Current research, future trends, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 668–695, 2021.

[2] A. H. Faeq, H. M. Nour, D. Kaharudin, H. E. Binti, S. Nurhizam, Q. Faizan, A. Khairul, and N. Q. Ngoc, "A survey on resource management for 6G heterogeneous networks: Current research, future trends, and challenges," *Electronics*, vol. 12, no. 3, 2023. [Online]. Available: https://doi.org/10.3390/electronics12030647

[3] A. Bharat, T. M. Amine, M. Marco, and M. Gabriel-Miro, "A comprehensive survey on radio resource management in 5G hetnets: Current solutions, future trends and open issues," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2495–2534, 2022. [Online]. Available: https://doi.org/10.1109/COMST.2022.3207967

[4] D. Ather, R. Kler, Z. T. Baig, G. P. Babu, A. Rastogi, and N. Ahmed, *6G Networks: Pioneering Advanced Communication Techniques for Call Centers and Beyond.* CRC Press, 2025. [Online]. Available: https://doi.org/10.1201/9781003583127-12

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004. [Online]. Available: https://web.stanford.edu/~boyd/cvxbook/

[6] A. Mughees, M. Tahir, M. A. Sheikh, A. Amphawan, Y. K. Meng, A. Ahad, and K. Chamran, "Energy-efficient joint resource allocation in 5G hetnet using multi-agent parameterized deep reinforcement learning," *Physical Communication*, vol. 61, p. 102206, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874490723002094

TABLE I: Quantitative comparison of DRL algorithms and heuristic baselines. The upward arrow (↑) indicates that a higher value is better, while the down arrow (↓) indicates a lower value is better. The best results are highlighted in bold, while the second-best results are indicated by underlining.

Scenarios	**Methods**	**Bandwidth Allocation (↑)**	**Transmit Power (↓)**	**Scheduling Score (↑)**
Dense Urban (10 N_S, 3 N_M, high interference)	G-OFDMA	0.09 ± 0.18	0.98 ± 0.01	0.03 ± 0.01
	IP-PC	0.21 ± 0.13	$\mathbf{0.01 \pm 0.03}$	0.17 ± 0.32
	PF-EQ	0.07 ± 0.08	0.29 ± 0.39	0.09 ± 0.12
	TD3	$\underline{0.57 \pm 0.14}$	0.40 ± 0.01	$\underline{0.51 \pm 0.01}$
	PPO	$\mathbf{0.72 \pm 0.01}$	$\underline{0.26 \pm 0.01}$	$\mathbf{0.65 \pm 0.01}$
Sparse Suburban (3 N_M only)	G-OFDMA	0.12 ± 0.10	0.82 ± 0.06	0.18 ± 0.08
	IP-PC	0.28 ± 0.09	$\mathbf{0.05 \pm 0.03}$	0.30 ± 0.10
	PF-EQ	0.15 ± 0.07	0.30 ± 0.09	0.35 ± 0.09
	TD3	$\underline{0.62 \pm 0.10}$	0.22 ± 0.03	$\underline{0.60 \pm 0.05}$
	PPO	$\mathbf{0.78 \pm 0.02}$	$\underline{0.15 \pm 0.02}$	$\mathbf{0.72 \pm 0.04}$
Hotspot (users cluster near N_S)	G-OFDMA	0.20 ± 0.14	0.85 ± 0.05	0.12 ± 0.06
	IP-PC	0.33 ± 0.11	$\underline{0.14 \pm 0.03}$	0.26 ± 0.12
	PF-EQ	0.18 ± 0.10	0.35 ± 0.12	0.28 ± 0.10
	TD3	$\underline{0.68 \pm 0.08}$	0.28 ± 0.03	$\underline{0.62 \pm 0.06}$
	PPO	$\mathbf{0.80 \pm 0.02}$	$\mathbf{0.14 \pm 0.01}$	$\mathbf{0.75 \pm 0.03}$
Mixed (random N_S + uniform users)	G-OFDMA	0.10 ± 0.12	0.90 ± 0.05	0.10 ± 0.05
	IP-PC	0.25 ± 0.10	$\underline{0.19 \pm 0.05}$	0.22 ± 0.10
	PF-EQ	0.12 ± 0.09	0.32 ± 0.10	0.24 ± 0.09
	TD3	$\underline{0.60 \pm 0.09}$	0.20 ± 0.01	$\underline{0.58 \pm 0.05}$
	PPO	$\mathbf{0.76 \pm 0.02}$	$\mathbf{0.16 \pm 0.02}$	$\mathbf{0.70 \pm 0.03}$

Algorithm 2 PPO for Resource Allocation Optimisation

```
Initialize actor-critic network parameters $\theta$.
for each iteration do
    Collect a set of trajectories by running policy $\pi_{\theta_{\text{old}}}$ in the environment for $T$ timesteps.
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$ (using GAE).

    for a fixed number of epochs do
        Optimise the surrogate objective on the collected data via stochastic gradient ascent: $\theta \leftarrow \theta + \alpha \nabla_\theta L^{\text{CLIP}}(\theta)$
    end for
    $\theta_{\text{old}} \leftarrow \theta$.
end for
```

TABLE II: Hyperparameters used for experiments

Parameters	**Values**
α	2.0×10^{-5}
λ	0.95
γ	0.99
ϵ	0.2
η	3.5
κ, β, ϕ	$1.0, 0.01, 0.96$

[7] M. Seli, B. P. Kumar, S. P. Kumar, B. S. Kishoro, H. K. Lee, and S. Mangal, "Mobility induced multi-hop leach protocol in heterogeneous mobile network," *IEEE Access*, vol. 10, pp. 132 895–132 907, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3228576

[8] Y. Shenghao, M. Jun, and L. Yanxiao, "Wireless network scheduling with discrete propagation delays: Theorems and algorithms," *IEEE Transactions on Information Theory*, vol. 70, no. 3, pp. 1852–1875, 2024. [Online]. Available: https://doi.org/10.1109/TIT.2023.3324180

[9] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglu, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *NIPS Deep Learning Workshop 2013*, 2013.

[11] van Hado Hasselt, G. Arthur, and S. David, "Deep reinforcement learning with double q-learning," ser. AAAI'16. AAAI Press, 2016, p. 2094–2100. [Online]. Available: https://doi.org/10.48550/arXiv.1509.06461

[12] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. H. J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," *arXiv preprint, arXiv:1812.05905v2*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1812.05905

[13] A. Archi, H. A. Saadi, and S. Mekaoui, "Applications of deep reinforcement learning in wireless networks-a recent review," in *2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)*, vol. 1, 2023, pp. 1–8.

[14] D. Tian, "An intelligent optimization method for wireless communication network resources based on reinforcement learning," *Journal of Physics: Conference Series*, 2023. [Online]. Available: https://doi.org/10.1088/1742-6596/2560/1/012036

[15] X. Chi, Z. Peifeng, Y. Haibin, and L. Yonghui, "D3qn-based multi-priority computation offloading for time-sensitive and interference-limited industrial wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 9, pp. 13 682–13 693, 2024. [Online]. Available: https://doi.org/10.1109/TVT.2024.3387567

[16] J. Park and W. Na, "Application of mac protocol reinforcement learning in wireless network environment," in *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, 2024, pp. 730–731.

[17] K. Olayemi, M. Van, S. McLoone, Y. Sun, J. Close, N. M. Nyat, and S. McIlvanna, "A twin delayed deep deterministic policy gradient algorithm for autonomous ground vehicle navigation via digital twin perception awareness," *arXiv preprint, arXiv:2403.15067v1*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2403.15067

[18] S. Shalini, N.Kopperundevi, R.Rajkumar, A. Radhika, M. Gopianand, and M. Ram, "Decentralized machine learning for dynamic resource optimization in wireless networks using reinforcement learning," *Journal of Electrical Systems*, 2024. [Online]. Available: https://doi.org/10.52783/jes.2539

[19] C. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[20] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *arXiv preprint, arxiv:9809099*, 1998.