Handover Optimization in 5G Networks Using Q-Learning (EC431 - 5G Communication and Network Report)

GitHub Code: https://github.com/IIITV-5G-and-Edge-Computing-Activity/Handover-Optimization-using-QLearning

Video Demo: https://youtu.be/wCoyhqpwW5o

Sanidhya Kumar
B.Tech Computer Science Engineering
202151138@iiitvadodara.ac.in

Riya
B.Tech Computer Science Engineering
202151132@iiitvadodara.ac.in

Samanway Maji
B.Tech Computer Science Engineering
202151136@iiitvadodara.ac.in

Shivang Bhargava
B.Tech Computer Science Engineering
202152339@iiitvadodara.ac.in

Anamika Sadh
B.Tech Computer Science Engineering
202151020@iiitvadodara.ac.in

***Abstract*—In a fully 5G network environment, gNodeBs, formerly referred to as antennas, are deployed in close proximity due to their limited coverage range. This results in a dense network topology, necessitating frequent reassignment of ongoing sessions from one gNodeB to another to ensure uninterrupted connectivity. This process, known as handover, can degrade network performance and lead to increased energy consumption. This study proposes optimizing handovers through reinforcement learning. A 3x6 grid-world environment simulating a dense network of closely spaced antennas is constructed. An agent navigates this environment randomly, gathering information about available antennas at each state. Using the Q-learning algorithm, a well-established method in reinforcement learning, a Q-table is developed over 5000 iterations. The antenna with the highest Q-value is then selected to minimize the frequency of handovers, thereby improving network efficiency and reducing energy costs.**

***Index Terms*—Handover, Q-learning, gNodeB, Reinforcement Learning, User Equipment.**

I. Introduction

The advent of Fifth-Generation (5G) networks marks a significant advancement in the field of wireless cellular technology, offering drastically increased speeds and the capacity to support a large number of simultaneously connected devices. In cellular systems, mobility is facilitated through the handover (HO) mechanism, which involves transferring an ongoing session from one gNodeB to a neighboring gNodeB, as illustrated in Fig. 1. Unlike Fourth-Generation (4G) networks, which feature relatively larger signal radii and infrequent handovers, 5G networks rely on numerous closely spaced gNodeBs to transmit signals. This results in a high frequency of inter-gNodeB handovers in standalone 5G deployments, potentially leading to connectivity issues. During a handover, user equipment temporarily disconnects from one gNodeB before establishing a connection with the next, causing disruptions in data connectivity. To address this challenge, non-standalone deployments combining 4G and 5G nodes have been utilized, with 4G nodes playing a dominant role. However, as standalone 5G environments are deployed, frequent handovers are inevitable, exacerbating issues such as data loss and the "ping-pong effect," where repeated handovers occur between two gNodeBs when user equipment is within their overlapping coverage areas. Given the absence of a pre-existing training dataset, reinforcement learning emerges as an effective solution. By allowing an agent to interact with the environment and gather information, we optimize the number of handovers. This approach minimizes connectivity disruptions and mitigates the ping-pong effect, ensuring seamless and efficient mobility in dense standalone 5G networks.

II. Formal problem and evaluation criteria

A Reinforcement Learning (RL) system primarily consists of five components: environment, agent, state, action, and reward. The RL framework is depicted in Fig. 2. To address the problem of providing a stable connection with minimal connectivity disruptions, we model the environment as a two-dimensional grid world representing a real-world path. The agent performs a random walk across the grid, navigating between any two points. The grid world includes multiple gNodeBs, ensuring that every coordinate point has access to signals from at least one gNodeB.

- State Space: Defined as the coordinate points in the grid world, e.g., (0,0), (2,0), etc.
- Action Space: Includes movements—right, left, up, and down, represented as [r, l, u, d].

The objective is to minimize connection disruptions by reducing the number of handovers. This is achieved through a reward mechanism: A reward of +1 is given when the agent remains connected to the same gNodeB. A penalty of -1 is assigned whenever a handover occurs. By following this reward function, the agent learns to avoid unnecessary handovers, maximizing rewards and ensuring stable connectivity.

III. Methodology

The proposed approach involves simulating a 5G grid environment to address the challenges of frequent handovers in dense 5G networks. A Reinforcement Learning (RL) agent navigates this grid from point A to point B, interacting with closely spaced gNodeBs. The objective is to minimize handovers by learning an optimal policy for antenna connections while maximizing Quality of Service (QoS) metrics such as signal strength and connection stability.

A. Simulation Environment

The environment is modeled as a 2D grid world representing a dense 5G network:

- Each cell represents a network area covered by an antenna.
- Multiple antennas ensure continuous signal availability, with overlapping coverage areas simulating a real-world small-cell network.
- The agent acts as the User Equipment (UE), like a smartphone, and moves randomly within the grid, selecting antennas based on signal availability.

B. Q-Learning Algorithm

Q-Learning, a value-based RL algorithm, is employed to optimize handover management. Key steps include:

- **Q-Table Creation**: A Q-Table records optimal actions for each state, iteratively updated using the Bellman equation.
- **Action Selection**: An epsilon-greedy strategy balances exploration and exploitation, enabling the agent to explore initially and exploit learned policies over time.
- **Reward Function**: A reward of +1 is given for avoiding handovers, while a penalty of -1 is assigned for each handover. This incentivizes stable connectivity.

C. Training Process

The training script runs the agent through multiple episodes, with:

- A grid size of 3×6 representing the network path.
- Each episode consisting of 150 steps, repeated for 5000 episodes.
- Performance metrics such as cumulative rewards and the number of handovers tracked to evaluate optimization.
- Epsilon decay applied progressively to prioritize exploitation as the training progresses.

D. Outcome and Evaluation

The optimized policy minimizes unnecessary handovers, evidenced by the Q-Table values and the reduction in handovers over time. This approach ensures efficient antenna selection and seamless mobility for the UE. The results demonstrate the potential of reinforcement learning to address dynamic challenges in 5G networks.

IV. Modules and Roles in 5G

The project consists of several interconnected modules, each playing a crucial role in simulating and optimizing handover management in a dense 5G network:

1) **GridWorld Environment (gridworld.py):**
   - Models the 5G network as a 2D grid where each cell is covered by an antenna.
   - Simulates overlapping signal coverage areas, replicating real-world small-cell network behavior.
   - Implements penalty and reward mechanisms to evaluate the agent's actions, penalizing unnecessary handovers and rewarding stable connections.
2) **Agent (agent.py):**
   - Represents the User Equipment (UE) navigating through the grid.
   - Uses Q-Learning to evaluate possible actions, such as connecting to a specific antenna, based on expected rewards.
   - Learns to minimize penalties and optimize handovers through exploration and exploitation strategies.
3) **Training Module (main.py):**
   - Integrates the agent with the GridWorld environment to run simulations.
   - Configures training parameters, including episodes, epsilon decay, and reward thresholds.
   - Logs performance metrics such as cumulative rewards, handovers, and steps per episode to monitor learning progress.
4) **Benchmarking Module (Q_learning.py):**
   - Compares the performance of the 5G-specific Q-Learning model to classical tasks like MountainCar.
   - Highlights the unique challenges of dense 5G networks, such as dynamic signal environments and frequent handovers.
   - Validates the robustness and adaptability of the proposed approach to real-world scenarios.
5) **Visualization Module (plotter.py):**
   - Generates visualizations of training results, including cumulative rewards, handovers per episode, and epsilon decay.
   - Provides histograms and trend analyses to summarize the agent's learning progress and optimization effectiveness.
   - Facilitates interpretation of results for further analysis and reporting.

V. Visualizing the Results

A. Cumulative Rewards over Episodes (Sampled)

This plot shows the cumulative rewards collected by the agent over 5000 episodes.

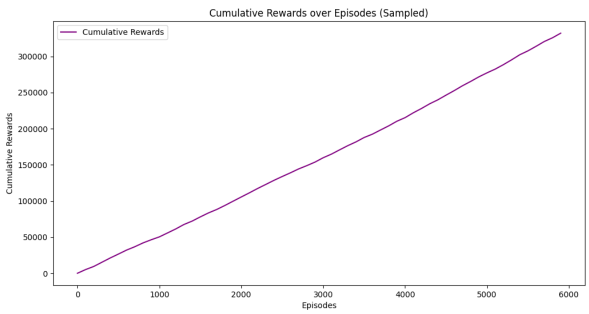

Fig. 1. Cumulative Reports (x-axis) vs Episodes (y-axis).

- The steady upward trend indicates that the agent is learning and improving over time.
- Rewards in this context represent the overall performance of the agent, where higher rewards signify fewer unnecessary handovers and better connectivity maintenance.
- A consistent increase without major dips suggests that the Q-Learning algorithm successfully trains the agent to make optimal decisions in the 5G environment.

B. Cumulative Success Rate

This plot tracks the cumulative success rate across episodes.

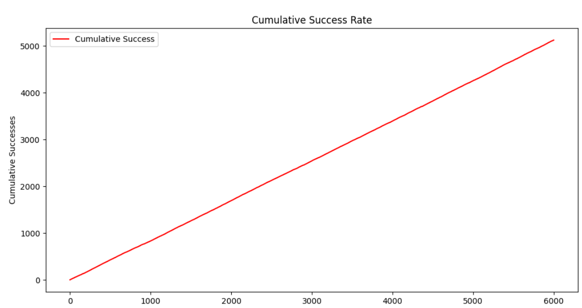

Fig. 2. Cumulative Success (x-axis) vs Episodes (y-axis).

- Success is likely defined as reaching the goal state (e.g., maintaining connectivity with minimal penalties) in each episode.
- The linear increase indicates consistent improvement, meaning the agent is learning to avoid errors (e.g., failed handovers) while completing its tasks efficiently.
- The high cumulative success rate near the end confirms that the agent has successfully generalized the optimal handover policy.

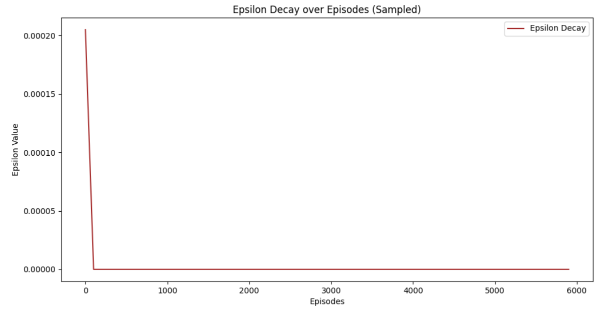

Fig. 3. Epsilon value (x-axis) vs Episodes (y-axis).

C. Epsilon Decay over Episodes (Sampled)

This plot displays the epsilon decay used to balance exploration and exploitation.

- At the start, a high epsilon value encourages the agent to explore the environment and try different actions, gathering data about state transitions and rewards.
- Over time, epsilon decreases, shifting the agent's behavior towards exploitation—using the knowledge it has gained to make optimal decisions.
- The rapid decay early in training reflects aggressive exploration, ensuring the agent learns efficiently before settling on a policy.

D. Handovers vs. Episodes (Sampled)

This plot shows the number of handovers per episode.

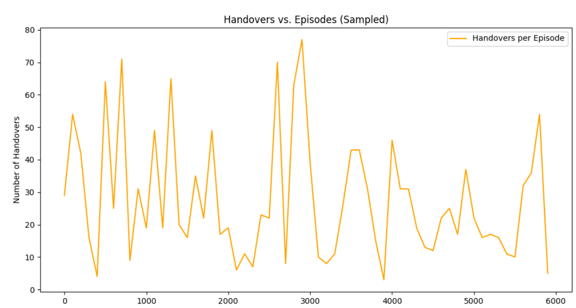

Fig. 4. Number of Episodes (x-axis) vs Episodes (y-axis).

- The fluctuations indicate varying environmental challenges and the agent's adaptive strategies.
- A downward trend or stabilization over time would suggest that the agent learns to minimize unnecessary handovers.
- Peaks in handovers may represent exploration phases or challenging episodes where the agent needs to adapt to dynamic network conditions.

E. Histogram of Handovers

This histogram shows the frequency distribution of handovers across all episodes.

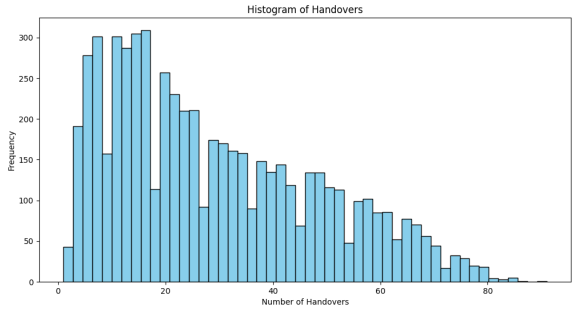

Fig. 5. Frequency (x-axis) vs Number of handovers (y-axis).

- The concentration of lower handover counts suggests that the agent has learned to minimize handovers effectively.
- A smaller frequency of high handover counts indicates that challenging scenarios occur less often as the agent becomes more proficient.
- The histogram provides a quick overview of the handover optimization success across the training process.

F. Rewards vs. Episodes (Sampled)

This plot shows the reward per episode along with a moving average.

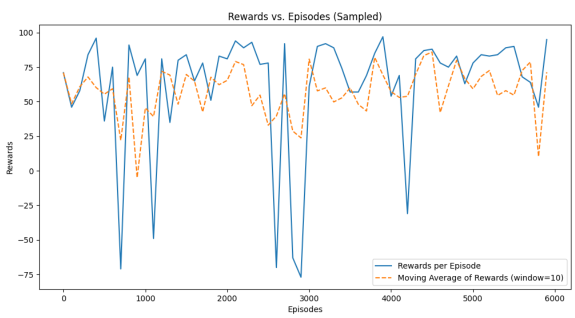

Fig. 6. Rewards (x-axis) vs Episodes (y-axis).

- The high variability in early episodes reflects the agent's exploration phase, where rewards fluctuate as it tries different strategies.
- The moving average smooths out short-term fluctuations and highlights long-term trends in the agent's performance.
- The increasing trend in the moving average demonstrates that the agent gradually learns better policies for managing handovers and maximizing rewards.

G. Steps Taken per Episode (Sampled)

This plot tracks the number of steps the agent takes in each episode.

- Fluctuations in step counts indicate varying levels of difficulty in navigating the environment.

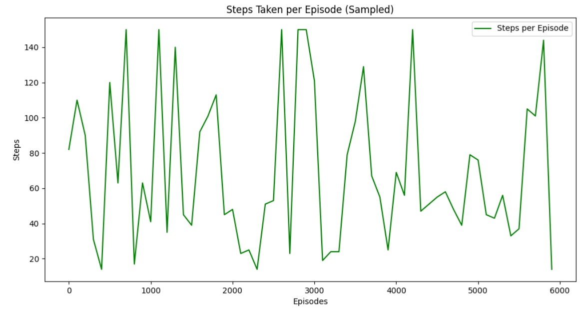

Fig. 7. Steps (x-axis) vs Episodes (y-axis).

- A gradual reduction or stabilization suggests the agent becomes more efficient in reaching its goal with fewer unnecessary movements.
- Peaks in steps may correspond to challenging scenarios or exploration phases.

H. Summary of Insights Across Plots

- **Learning Efficiency:** The cumulative rewards and success rate plots confirm that the agent effectively learns and improves over time.
- **Optimization of Handovers:** The handovers-related plots (histogram and episodes plot) validate that the agent reduces unnecessary handovers, which aligns with the project's goals in a 5G network.
- **Exploration vs. Exploitation:** The epsilon decay plot ensures that the agent balances exploration and exploitation effectively.
- **Overall Performance:** The steps and rewards plots demonstrate that the agent not only learns optimal policies but also executes them efficiently with minimal resource usage.

These plots collectively showcase the robustness of the Q-Learning approach in optimizing handover management in a simulated 5G environment.

Metric	Benchmark Task (MountainCar)	5G Task (Handover Optimization)
Convergence Speed	Fast (e.g., ~500 episodes)	Slow (e.g., ~3000 episodes)
Cumulative Rewards	High and consistent	Lower with more variability
Steps Taken per Episode	Optimized quickly	Requires more steps initially
Complexity (State Space)	Small	Large

Fig. 8. Comparison table

VI. Project Bifurcation

The project responsibilities and deliverables were distributed among team members to ensure efficient execution while leveraging individual strengths. The bifurcation is detailed as follows:

A. Team Member 1: Shivang

Responsibilities:

- Implement and fine-tune the `gridworld.py` script to model the 5G network grid environment.
- Define antenna placements, signal coverage, penalties for handovers, and rewards for optimal behavior.
- Test the environment for edge cases and ensure it supports dynamic updates like varying signal strengths.

Deliverables:

- A functional grid environment with all parameters and behaviors defined.
- A set of test cases ensuring proper state transitions and reward/penalty mechanisms.

B. Team Member 2: Sanidhya

Responsibilities:

- Implement the `agent.py` script with the Q-Learning algorithm for handover optimization.
- Develop functions for action selection, antenna selection, Q-value updates, and epsilon decay.
- Integrate the agent with the GridWorld environment and ensure compatibility.

Deliverables:

- A fully functional agent capable of learning optimal policies.
- Documentation of Q-Learning parameters and their impact on training performance.

C. Team Member 3: Samanway

Responsibilities:

- Develop and manage the `main.py` script to set up the environment and agent for training.
- Handle the training loop, logging metrics like rewards, handovers, and steps per episode.
- Integrate epsilon decay logic and ensure smooth progression from exploration to exploitation.

Deliverables:

- A script that successfully trains the agent over multiple episodes.
- Logs or data files recording performance metrics for further analysis.

D. Team Member 4: Riya

Responsibilities:

- Develop the `plotter.py` script to generate performance plots (e.g., cumulative rewards, handovers per episode, epsilon decay).
- Analyze the plots to assess agent performance and identify areas for improvement.
- Create histograms and moving averages to summarize the results.

Deliverables:

- Clear, informative plots saved in the `metrics_plots/` directory.
- A report summarizing the findings from the visualizations and any recommendations for improvement.

E. Team Member 5: Anamika

Responsibilities:

- Implement and manage the `Q_learning.py` script for benchmarking the Q-Learning algorithm on a simpler task (e.g., MountainCar).
- Compare the results from the benchmark task with the main 5G handover optimization task.
- Document key insights from the comparison and highlight the unique challenges of the 5G environment.

Deliverables:

- A functional Q-Learning benchmark script with clear outputs.
- A comparative analysis report highlighting the differences in learning efficiency, rewards, and challenges.

F. Collaborative Tasks

- **Integration Testing:** All members worked together to test the integration of the environment, agent, and training script.
- **Final Presentation:** A joint presentation was created to summarize the project goals, methods, results, and future work. Each member presented their respective sections.

This distribution ensured that each member contributed significantly to the project while focusing on their strengths and responsibilities.

VII. Conclusion

This project addresses some of the key challenges in 5G networks, showcasing the effectiveness of Q-Learning in optimizing handover management. The major challenges and their solutions are summarized below:

A. Key 5G Challenges Addressed

- **Dynamic Antenna Selection:** The agent dynamically chooses the best antenna based on its position and signal strength, ensuring minimal latency and high throughput.
- **Seamless Handover Management:** By penalizing unnecessary handovers, the agent learns to switch antennas only when necessary, maintaining stable connectivity.
- **Real-Time Adaptation:** The Q-Learning framework enables the agent to adapt to changing signal conditions, simulating real-world network dynamics.
- **Scalability:** The grid environment can easily be scaled to include more antennas, users, or more complex handover scenarios.

B. Future Applications in 5G

This project lays the foundation for advanced research and development in 5G networks. Potential future applications include:

- **Multi-Agent Systems:** Collaborating agents for load balancing and resource allocation in dense networks.
- **Deep Reinforcement Learning:** Using neural networks to handle larger, more complex 5G environments.
- **Real-World Deployment:** Integrating live network data to optimize handovers in operational 5G systems.

C. Summary

In summary, this project demonstrates how reinforcement learning, specifically Q-Learning, can be applied to optimize handover management in 5G networks. It provides a framework for tackling real-world challenges such as dynamic antenna selection, seamless connectivity, and adaptive learning in complex network environments.

The insights and methodologies presented here pave the way for scalable, efficient, and adaptive solutions to improve connectivity in dense 5G deployments, contributing to the future of telecommunications.

References

1) V. Yajnanarayana, H. Ryden, and L. Hévizi, "5G handover using reinforcement learning," in *2020 IEEE 3rd 5G World Forum (5GWF)*, pp. 349–354, 2020.
2) A. Masri, T. Veijalainen, H. Martikainen, S. Mwanje, J. Ali-Tolppa, and M. Kajo, "Machine-learning-based predictive handover," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 648–652, 2021.
3) X. Cai, C. Wu, J. Sheng, J. Zhang, and Y. Wang, "A parameter optimization method for LTE-R handover based on reinforcement learning," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1216–1221, 2020.
4) A. Alkhateeb, I. Beltagy, and S. Alex, "Machine learning for reliable mmWave systems: Blockage prediction and proactive handoff," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1055–1059, 2018.