Reinforcement Learning Literature Survey:
Handover Optimization in Wireless Networks for Mobile Users and Base Stations

Alperen Duru and Rosemary Lach

Summary of Plans and Progress

The plan for this project has not changed significantly since the project proposal. The primary goal is still to expand upon [1] by applying function approximation methods to the handover optimization problem outlined in the paper and comparing these results to those of deep reinforcement learning. Efforts so far have been to understand the existing code framework, and to generate our own dataset with QuaDRiGa that is compatible with this framework, as well as the additional function approximation code that will be produced later. Code for this project can be found at https://github.com/rosemarylach/RL-Project-LB.git and contains all of the work generated thus far. The moving BSs scenario is going to be implemented if time permits. Moreover, a simplified channel with only large-scale fading and path loss will be used for the moving BSs scenario since the actions directly affect the received powers compared to the original static case.

Literature Survey

There are many opportunities for reinforcement learning to be used in the field of wireless communications, as many network systems can be modeled as a Markov Decision Process (MDP), where entities need to make decisions to optimize some network performance parameters. Deep reinforcement learning has been of particular interest in multiple problems such as traffic routing and resource allocation [2].

This project is primarily centered around the fundamental ideas presented in [1], a paper that focuses on load balancing and handover optimization using deep reinforcement learning. Gupta sets up a sequential decision-making problem that attempts to maximize the sum-log-rate as its reward by adaptively finding UE-BS associations with parameter $x_{ij}[t]$ for UE i and BS j. The state space consists of UE locations, velocities, and loads on the system. This project will use this same formulation and state space while attempting to use function approximation to solve the RL problem. Expanding this project further to include mobile BSs would also change the state and action spaces to account for the BS position. Due to the proprietary nature of the dataset used in [1], ray tracing simulations must be used to develop our own dataset of channel matrices that can be used to test both the original deep reinforcement learning methods, as well as new function approximation methods. New 3GPP standards as outlined in [3] will be used for this dataset, as opposed to the version published in 2017 that was originally used. However, no standards that are relevant to this project have changed since then.

To generate the dataset, we use QuadRiGa, a simulation software that generates channel matrices via ray tracing in a custom environment [4]. This is the same tool used in the original paper, however the environment setup is unique to this project. We attempt to replicate the original environment as best as possible using the documentation provided in [5] and general information [1] did provide, but we recognize that there may be some differences. For static BSs, a set of reference signal received powers (RSRPs) can be generated and then the individual signal-to-interference and noise ratio (SINR) can be computed for each UE-BS association as done in [1]. For the replication of the work in [1], we can produce a single episode of moving UEs' RSRPs and can learn a policy by changing the current UE-BS associations, much similar to the planning introduced in [6]. Generating more episodes should help avoid overfitting to a particular episode of realization. However, when the BSs are also moving, the actions taken by the agent will affect the next positions of the BSs, affecting the next set of RSRPs. For this reason, QuadRiGa is not useful for the dataset generation for the moving BSs scenario and a simplified channel with large-scale fading and path loss will be simulated in Python, together with the learning environment.

In the literature, similar states and rewarding mechanisms taking into account the handover timings exist with different learning approaches to a similar problem. Authors of [7] discuss the mobile UAV and mobile users association for a mobile edge computing scenario and solve the learning problem by using the twin-delayed deep deterministic policy gradient (TD3). However, they only have 1 moving BS terminal and they are optimizing for the UAV trajectory only. Authors of [8] use deep neural networks to do the learning asynchronously at the UE side by clustering some UEs for faster and better learning. Multi-armed bandits and contextual bandits approaches are tested in [9] and [10], respectively, where [10] also includes the beamforming aspects of the system. Finally, [11] proposes a deep Q-learning approach to optimize for energy efficiency, where we have a different reward metric of sum-log-rate. Our work is going to distinguish itself from the literature by building on top of [1] and then changing the learning methods to include different function approximation methods, potentially adding the moving BSs scenario if time permits. The expected deliverables of the project are to compare different learning techniques' learning speed and overall performance in graphs and if time permits, also add the moving BSs scenario as an extra result.

Sources

[1] M. Gupta *et al.*, "Load balancing and handover optimization in multi-band networks using Deep Reinforcement Learning," *2021 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2021. doi:10.1109/globecom46510.2021.9685781

[2] N. C. Luong *et al.*, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133-3174, Fourthquarter 2019, doi: 10.1109/COMST.2019.2916583.

[3] *Study on channel model for frequencies from 0.5 to 100 GHz (Release 17)*, document 3GPP TR 38.901, Jan. 2024.

[4] S. Jaeckel, L. Raschkowski, K. Börner and L. Thiele, "QuaDRiGa: A 3-D Multi-Cell Channel Model With Time Evolution for Enabling Virtual Field Trials," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 3242-3256, June 2014, doi: 10.1109/TAP.2014.2310220.

[5] S. Jaeckel, L. Raschkowski and L. Thiele, *Quasi Deterministic Radio Channel Generator User Manual and Documentation (Revision v2.8.1)*, Dec. 2023.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Massachusetts: MIT Press Ltd, 2018.

[7] S. Hwang, J. Park, H. Lee, M. Kim and I. Lee, "Deep Reinforcement Learning Approach for UAV-Assisted Mobile Edge Computing Networks," *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, Rio de Janeiro, Brazil, 2022, pp. 3839-3844, doi: 10.1109/GLOBECOM48099.2022.10001011.

[8] Z. Wang, L. Li, Y. Xu, H. Tian and S. Cui, "Handover Control in Wireless Systems via Asynchronous Multiuser Deep Reinforcement Learning," in *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4296-4307, Dec. 2018, doi: 10.1109/JIOT.2018.2848295.

[9] Y. Zhao and X. Luo, “Handover mitigation in dense hetnets via bandit arm elimination,” *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019. doi:10.1109/globecom38437.2019.9013926.

[10] V. Yajnanarayana, H. Ryden, and L. Hevizi, “5G handover using reinforcement learning,” *2020 IEEE 3rd 5G World Forum (5GWF)*, Sep. 2020. doi:10.1109/5gwf49715.2020.9221072.

[11] Y. Song, S. H. Lim, and S.-W. Jeon, “Distributed online handover decisions for energy efficiency in dense HetNets,” *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec. 2020. doi:10.1109/globecom42002.2020.9348215.