Power Optimization in Device-To-Device Communications: A Deep Reinforcement Learning Approach with Dynamic Reward

Jiyun Ha, Jamie Shroufe

INTRODUCTION

The paper *"Power Optimization in Device-to-Device Communications: A Deep Reinforcement Learning Approach with Dynamic Reward"* addresses improving the efficiency of device-to-device (D2D) communication in modern wireless networks. D2D communication, a key feature of 5G, allows devices to communicate directly with each other, leading to faster connections, better network capacity, and reduced congestion on cellular networks. However, a significant challenge lies in managing interference between devices, which can impact network performance (throughput) and energy efficiency (EE).

To tackle this, the authors proposed a novel method using deep reinforcement learning (RL) to optimize power allocation dynamically. Their approach involves two parallel deep Q-networks (DQNs), each focusing on a specific objective: maximizing data throughput or improving energy efficiency. By dynamically adjusting rewards based on the network's current state, the algorithm can effectively balance these objectives while meeting quality-of-service (QoS) requirements for both D2D users and traditional cellular users.

Ideally, the proposed algorithm shows improved energy efficiency and system performance compared to traditional methods, making it a promising approach for the future of communication technology.

COMPLICATIONS

However, replicating this project posed significant challenges, primarily due to its high computational demands. The deep reinforcement learning algorithm introduced in the paper requires extensive training on large datasets, with numerous iterations to optimize the two deep Q-networks (DQNs).

Each DQN involves complex computations for evaluating state-action pairs and updating weights, which demand significant processing power and memory. Not to mention, the training process in the paper involves thousands of episodes to fine-tune the model, with each episode requiring multiple simulations of network states and actions.

To achieve the same level of results as presented by the authors, we would have likely needed to dedicate our laptops exclusively to running the algorithm for several weeks. This was, unfortunately, impractical given the limited timeframe of our project and also the relatively low computational power of our personal laptops. Unlike the high-performance hardware typically accessible to research labs, we struggled to handle the computational load required by the deep reinforcement learning model.

As previously mentioned, the training process involves thousands of iterations, with each step requiring complex matrix calculations, reward evaluations, and neural network updates. As a result, we frequently encountered glitches, system slowdowns, and outright crashes when attempting to run the program for extended periods.

Even with adjustments to reduce computational intensity, such as limiting the network size and significantly decreasing the number of training episodes, the program remained highly resource intensive. The lack of a dedicated GPU or even just sufficient RAM in our laptops to process the large datasets efficiently was a major obstacle as it resulted in insanely long runtimes and inconsistent program performance.

RESULTS

Nonetheless, our results are as follows. Although we had to significantly reduce the scope of our project down to just a few training episodes and even fewer simulations per episode, Figure 1 depicts our resulting training loss for both the EE and the throughput.

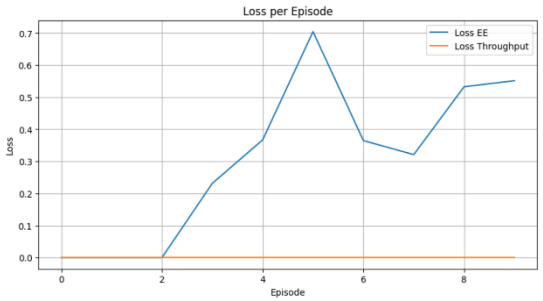

Figure 1. Our Graph Depicting Training Loss

This can be compared to Figure 2 which depicts the ideal training loss for EE and throughput (as published in the original paper).

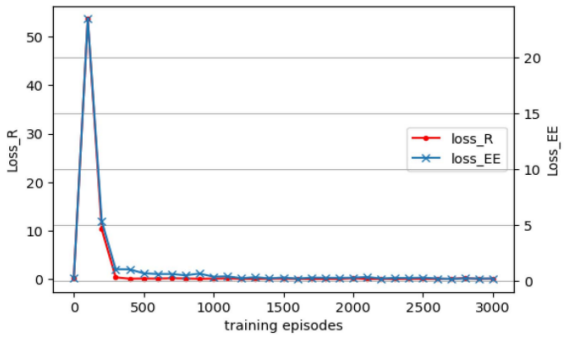

Figure 2. Ideal Graph Depicting Training Loss (Taken from Paper)

Unfortunately, the graph we produced and that presented by the original authors are visibly different. This discrepancy is largely due to the simplifications we made in our calculations that reduced the number of simulations and training episodes. These adjustments, while necessary given our computational limitations, mean that we were unable to replicate the exact results achieved in the original study.

While it is challenging to directly compare the two graphs due to the distinct scaling and altered parameters in our implementation, this does not necessarily imply that our graph is incorrect. The difference in scaling and experimental conditions makes it likely that the underlying trends and relationships captured in our graph still hold validity, even if they do not match the original precisely.

The same can be said for our graph pertaining to training rewards (Figure 3).

Figure 3. Our Graph Depicting Training Reward.

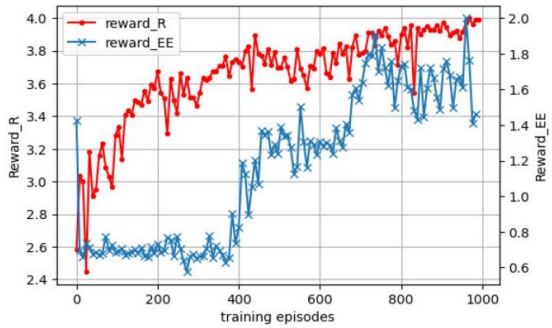

Figure 4. Ideal Graph Depicting Training Reward (Taken from Paper)

Had we obtained the same results as the authors, we would have observed a similar trend in the training loss displaying a rapid increase followed by a sharp decrease, ultimately leveling off to a near-zero value for both EE and throughput. This pattern would suggest that the model effectively learned the optimal policies for minimizing loss, demonstrating its capability to adapt and perform in a manner consistent with the intended outcome. The convergence toward a near-zero value would be a strong indicator that the model successfully optimized its performance metrics, aligning with the findings reported by the authors.

Additionally, we would likely see similar trends in the training rewards. For throughput rewards, we would expect to see a sharp increase in the first 200 episodes, followed by a slower but steady rise in the following episodes, eventually stabilizing around 3.9 to 4.0. In contrast, EE rewards would likely remain flat for the first 400 episodes before jumping sharply between episodes 400 and 700. After that, the reward would start to fluctuate but would eventually stabilize around 1.8 to 2.0. These stabilizing trends show how the model gradually learns to improve throughput and energy efficiency over time.

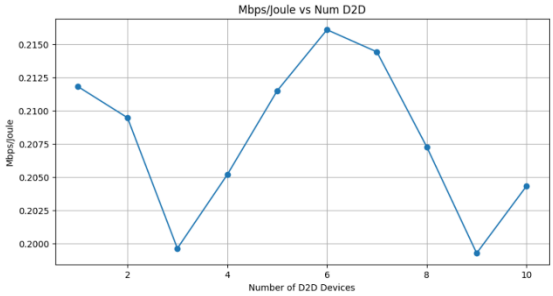

Figure 5. Our EE Performance Graph

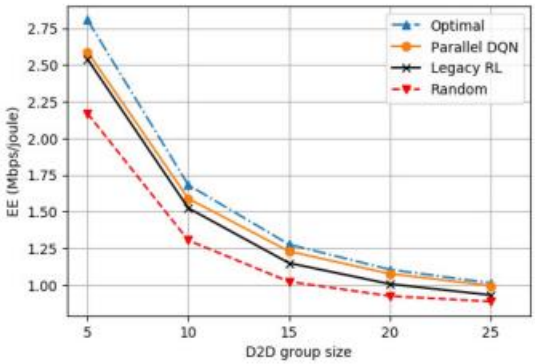

Figure 6. Ideal EE Performance Compared to the Optimal Performance, Legacy RL (Previous Model) Performance, and Random Performance (Taken from Paper)

Lastly, Figures 5 and 6 serve to compare the EE performance. The authors' graph shows four different performance benchmarks, while ours only shows one. This is due to the fact that adding more

reinforcement learning algorithms would have just been too much as we would have run into the same processing issues, just for more algorithms.

While the trends in the two graphs do not align perfectly, this can be attributed to the fact that we had to significantly reduce the scope of our project. As a result, we were unable to achieve the expected EE regression that we had hoped for, producing different performance outcomes as the original study.

CONCLUSION

Overall, this study on power optimization in D2D communication using deep RL offers a promising approach to enhance network efficiency. The authors' algorithm of using two parallel deep Q-networks (DQNs) to optimize both throughput and energy efficiency (EE) provides a balanced tradeoff for improving system performance. By adjusting rewards based on the network's real-time state, their algorithm ensures quality-of-service (QoS) for both D2D and traditional cellular users.

However, as mentioned, replicating the study was difficult due to the high computational demands of the deep RL model. Our laptops lacked the necessary processing power, so we had to significantly simplify the experiment by reducing the number of episodes and simulations. While this led to some differences in results, we are hopeful that had we ascertained sufficient processing power, we would have been able to produce similar results.

Looking ahead, there are several areas where this approach could be expanded and improved. Incorporating additional QoS metrics, such as application performance and service reliability, could provide a more comprehensive understanding of user satisfaction. Additionally, extending the algorithm's application beyond 5G networks to even more densely populated environments would also be valuable, as the dynamics of power allocation differ in highly congested networks. Similarly, comparing the algorithm's performance in urban versus rural areas could reveal insights into how environmental factors influence its effectiveness. Finally, real-world testing of this approach would be crucial to validate its practicality and performance outside of controlled simulations.

While our results did not fully replicate the original study due to computational limitations, they still highlight the potential of deep RL in optimizing power allocation for wireless communication systems. This paves the way for future research in this field and demonstrates the importance of continuous improvement and testing as we move towards more advanced network architectures.