CHAPTER 1

INTRODUCTION

1.1 Background

Mobile Edge Computing (MEC) has become a cornerstone in the evolution of modern mobile networks and emerging technologies by extending cloud computing capabilities closer to end-users [1, 2]. This proximity not only reduces latency but also enhances the performance of mobile devices, optimizes bandwidth by minimizing data transmission to centralized servers, and bolsters security and privacy through localized data processing [3]. MEC is crucial in supporting real-time applications, including augmented reality, autonomous driving, and industrial IoT, where low latency and high reliability are imperative.

Globally, the exponential growth of data-intensive applications and connected devices has driven the demand for MEC solutions that can effectively manage the massive computational and communication requirements at the network's edge. However, MEC environments' inherent dynamic and resource-constrained nature poses significant challenges, particularly in optimizing offloading decisions and resource allocation [4]. Variability in network conditions, such as fluctuating channel gains and varying power levels, necessitates the development of adaptive algorithms capable of efficiently managing these resources in real time. The ability to dynamically allocate tasks between local devices and edge servers is crucial for maintaining the quality of service (QoS) in these rapidly evolving environments.

Deep Reinforcement Learning (DRL) has emerged as a promising approach to tackling these challenges by enabling MEC systems to learn optimal policies for task offloading and resource allocation through continuous interaction with the environment [1, 5]. DRL-based methods, such as the DROO algorithm, have demonstrated considerable potential in enhancing decision-making processes by adapting to changing network conditions and improving overall system performance [6]. Despite these advancements, the computational complexity and time efficiency of DRL algorithms remain a critical concern, especially in large-scale MEC deployments where real-time decision-making is essential [4].

The current state of real-time offloading simulations reveals that conventional optimization methods often fail to resolve complex combinatorial problems swiftly enough to cope with the rapidly changing wireless channel conditions inherent in MEC environments [1, 5]. Traditional optimization algorithms, such as Adam, have been widely utilized in training deep learning models due to their ability to handle sparse gradients and achieve fast convergence. However, emerging variants like Nadam, which incorporate Nesterov momentum, promise further improvements in convergence speed and stability, though they also introduce additional computational overhead [7].

The primary challenge lies in setting adaptive thresholds for efficient task offloading while considering the variations in network conditions and workloads [8]. There is a pressing need for algorithms that can make efficient, real-time computation offloading decisions and dynamically adapt to fluctuating channel conditions [1, 5]. This research addresses these issues by exploring advanced optimization techniques, including deep reinforcement learning, which offer potential solutions through their ability to learn optimal offloading strategies in dynamic network environments.

By leveraging these intelligent algorithms, the study aims to enhance the adaptability and efficiency of MEC systems, overcoming the limitations of conventional methods and fulfilling the stringent demands of modern mobile networks and IoT applications. The expected outcome is the development of more resilient and responsive MEC architectures capable of supporting a broad range of real-time and computationally intensive applications, ultimately leading to improved user experiences and enhanced network performance.

1.2 Problem Identification

Mobile Edge Computing (MEC) has emerged as a vital framework in modern networks, particularly in scenarios where low latency, high bandwidth, and real-time processing are paramount. The increasing demand for data-intensive applications, such as augmented reality, autonomous driving, and industrial IoT, has driven the adoption of MEC to bring computation closer to the data source, thus reducing latency and improving user experience [3]. However, this shift introduces a set of challenges that need to be addressed to realize the full potential of MEC.

One of the critical challenges in MEC is the optimization of computation offloading decisions, where tasks are dynamically allocated between local devices and edge servers. The dynamic nature of wireless networks, characterized by fluctuating channel conditions, varying device power levels, and limited computational resources, makes the offloading decision process complex and computationally intensive [4]. Efficiently managing these resources in real-time is crucial for maintaining the quality of service (QoS) and ensuring that MEC systems can meet the stringent requirements of modern applications [9].

The deployment of DL services utilizes local resources situated at the edge of the network, resulting in enhanced efficiency and diminished latency [10]. The time delay can be reduced by processing data close to its origin. Furthermore, the utilization of local computing resources leads to improved efficiency. Processing data at its source contributes to a reduced response time, which is of paramount importance for real-time applications [8]. The employment of edge computing obviates the need for dispatching data to remote cloud servers, thereby enabling expedited and more efficient operations.

Traditional optimization techniques, such as convex optimization, have been applied to solve offloading problems in MEC, but these methods often struggle with scalability and adaptability in highly dynamic environments. Deep Reinforcement Learning (DRL) has been proposed as a solution to this problem due to its ability to learn and adapt to complex, dynamic environments through continuous interaction with the system [1]. However, DRL-based approaches also introduce new challenges, particularly related to the computational complexity and time required for training these models. The efficiency of the training process, therefore, becomes a significant bottleneck, especially in large-scale MEC deployments [10].

To address these challenges, this research explores the application of various DRL optimization algorithms to improve the efficiency and stability of offloading decisions in MEC environments. Previous studies have demonstrated the potential of different algorithms, such as Adam, in achieving fast convergence [11]. However, the performance of these algorithms in highly dynamic and resource-constrained environments like MEC remains an open area of investigation. This study will systematically evaluate the performance of these algorithms under varying network conditions to identify an approach that can offer a balance between computational efficiency and robust performance.

Furthermore, bisection optimization is used to enhance Q-value by bestowing rewards to the DRO model processing outcomes to attain superior decision results in each iteration [1]. For resource allocation, the optimal time allocation of the convex problem (P2) can be efficiently solved using a one-dimensional bi-section search over the dual variable associated with the time allocation constraint in O(N) complexity [12].

In previous research [1], the utilization of 30 User Wireless Devices (WDs) successfully optimized CPU latency, resulting in a time interval of less than 0.1 seconds. These simulations were conducted on a desktop equipped with an Intel Core i5-4570 3.2 GHz CPU and 12 GB of memory [1]. Interestingly, when the experiment was repeated with only 10 User WDs, an even more significant reduction in latency was observed, specifically maintaining the time interval at 0.1 seconds. This finding highlights a potential area for further research, suggesting that there may be opportunities to enhance system performance by optimizing the number of WDs concerning CPU latency. By comparing these results with those obtained using various DRL algorithms and other methodologies, this research can contribute to a deeper understanding of how different configurations impact system performance in MEC environments.

1.3 Objective and Hypotheses

The objectives of this research are as follows:

1. Optimization of Offloading Decisions using Deep Reinforcement Learning. Propose and implement a deep reinforcement learning-based online offloading framework to optimize offloading decisions in the context of Mobile Edge Computing (MEC). Specifically, evaluate and compare the performance of the Adam, Nadam, and My-

Nadam algorithms in terms of computation time, offloading efficiency, and overall system performance.

2. Integration of Deep Learning in Edge Computing. Integrate Deep Learning methodologies into edge computing scenarios, with a focus on enhancing efficiency, reducing latency, and improving decision-making processes. Investigate the deployment of Deep Learning models at the edge of the network to address the unique challenges posed by the dynamic and resource-constrained environment of MEC.

3. Investigation and Evaluation of Optimization Algorithms in Deep Reinforcement Learning. Investigate the effectiveness of different optimization algorithms (Adam, Nadam, and MyNadam) in deep reinforcement learning for MEC. Specifically, assess their performance in terms of convergence speed, stability under varying load conditions, and computational efficiency, particularly in large-scale MEC deployments.

Hypotheses:

- *H1:* The MyNadam algorithm will demonstrate superior computational efficiency and reduced latency compared to Adam and Nadam in the context of MEC, particularly under varying load conditions.

- *H2:* Integrating Deep Learning at the edge of the network will result in significant improvements in real-time decision-making processes and overall system performance in MEC environments.

- *H3:* Deep reinforcement learning algorithms, when optimized using MyNadam, will show enhanced resilience to gradient scaling issues and will perform more effectively in large-scale MEC scenarios.

1.4 Scop and Delimitation

This study focuses on the development and evaluation of a modified version of the Nadam optimizer, termed MyNadam, specifically designed for use in Mobile Edge Computing (MEC) environments. The scope of this research is limited to the following aspects:

1. **Algorithms and Optimization:**
   - The study evaluates the performance of several deep reinforcement learning (DRL) optimization algorithms, including Adam, Adadelta, Adagrad, Adamax, Nadam, and the newly developed MyNadam.
   - The modification of Nadam involves the removal of the *u_product* component to reduce computational complexity while maintaining or improving performance.
2. **Mobile Edge Computing (MEC) Environment:**

- The experiments are conducted within simulated MEC environments, focusing on optimizing task offloading decisions under varying network conditions, including normal and alternate (increased) load scenarios.
- The research does not extend to real-world implementations or testing in physical MEC networks but relies on simulated data and conditions to approximate real-world scenarios.

3. **Performance Metrics:**
   - The study measures performance based on several key metrics: Normalized Computation Rate, Total Time Consumed, Average Time Per Channel, and Training Loss.
   - These metrics are used to compare the efficiency and effectiveness of MyNadam against traditional optimizers like Adam and Nadam with *u_product*.

4. **Dataset and Simulation Conditions:**
   - The experiments use datasets that simulate various network conditions typical of MEC environments. These datasets include parameters like channel gain, transmission power, and computation rate, which are critical for task-offloading decisions.
   - The simulation involves testing with different iteration counts (e.g., 10,000, 5,000, 4,000, and 2,000) to observe the impact of MyNadam across various computational loads and environments.

5. **Generalization and Applicability:**
   - While the findings from this research provide insights into the performance improvements achievable with MyNadam, the results are specific to the simulated environments and configurations used in this study.
   - The applicability of MyNadam to other types of networks or optimization problems outside MEC is not covered in this research and would require further investigation.

6. **Delimitations:**
   - The study does not explore other potential modifications to Nadam or compare MyNadam with optimizers outside the set tested (e.g., RMSprop, SGD).
   - It also does not consider the impact of MyNadam on different types of neural network architectures beyond those used in the experiments, nor does it delve into hardware-specific optimizations or parallel computing frameworks.

- The research is confined to a controlled simulation environment, and results may vary when applied to actual MEC deployments.

By focusing on these specific areas, the study aims to contribute to the optimization of DRL-based computation offloading in MEC systems, while acknowledging the controlled scope and the limitations inherent in simulated testing environments.