Opti DRL: Integrating Optical Networks With Multi-Objective Optimization And Deep Reinforcement Learning For MEC Resource Allocation

Akshita Chaudhary[1], Pritibha Sukhroop[2], Amit Sharma[3], Mridula Bhardwaj[4], Barkha Samania[5]

[1]Deptt.of M.C.A. SRM Institute of Science & Technology NCR Campus
[2]Deptt. of EE, ABSS Institute of Technology
[3]Deptt. of CSE, SCRIET C.C.S.University
[4]Deptt.of CSE, HLM Group of Institutions,Ghaziabad
[5]Deptt.of M.C.A. SRM Institute of Science & Technology NCR Campus
[2]Corresponding Author- pritibhadhadli@gmail.com Pritibha Sukhroop

Abstract:

The sudden boom of latency-demanding and bandwidth-hungry applications in 5G and beyond has loaded tremendous pressure upon Mobile Edge Computing (MEC) infrastructures. Resource provisioning is important in order to meet the stringent quality-of-service (QoS) demands, while conventional approaches do not perform well with the dynamism and heterogeneity of the MEC setting. Furthermore, current solutions largely ignore the strength of high-speed optical networks as a means for improving MEC performance. We introduce OptiDRL, a new framework that combines optical network infrastructure with deep reinforcement learning (DRL) and multi-objective optimization to enable intelligent and adaptive resource allocation in MEC systems in this paper.

OptiDRL casts MEC resource allocation as a multi-objective decision-making problem, balancing latency, energy usage, and resource utilization. A DRL agent is learned in this context with a well-crafted reward function that balances these objectives. The optical integration provides ultra-low latency and high-throughput communication, further improving system efficiency. We deploy and test OptiDRL on simulation environments mimicking real-world MEC environments.

Experimental results show that OptiDRL outperforms current state-of-the-art benchmark algorithms with a significant latency reduction of up to 35%, resource saving of 25%, and scalability improvement under changing workload scenarios. This paper proves the potential in integrating optical networking with DRL to advance intelligent MEC resource management.

Keywords: MEC system, QOS, OptiDRL, multi-Objective optimization, optical network infrastructure, Decision making problems, Adaptive resource allocation, Throughput, Latency, Optical networking.

1. Introduction

The explosive growth of wireless technologies and the spread of latency-sensitive applications—like augmented reality (AR), virtual reality (VR), autonomous driving, and smart healthcare—have put an unprecedented burden on network performance. In order to counteract such demands, Mobile Edge Computing (MEC) has come forward as a promising paradigm through the facilitation of computation and storage facilities in proximity to end users, thus drastically lowering latency and enhancing service responsiveness. Nonetheless, the distributed and dynamic character of MEC poses new challenges, such as in resource allocation, in which computational, storage, and bandwidth resources should be efficiently handled in order to optimize performance and QoS.

The common approach to traditional resource allocation mechanisms in MEC relies on heuristic algorithms, linear optimization methods, or static rules. Though such approaches can hold up in static or deterministic environments, they are not adaptable and scalable in response to actual-time workload fluctuations and intricate network dynamics. Moreover, they tend to concentrate on one optimization goal, e.g., latency minimization or throughput maximization, while neglecting the inherently multi-objective nature of MEC systems where trade-offs among several performance metrics—e.g., energy consumption, delay, and resource utilization—need to be addressed.

Another essential area that is not well explored in MEC research is the convergence of high-capacity communication infrastructures, especially optical networks. Optical networking, being high bandwidth and low latency, can greatly contribute to improving MEC's data transmission and avoiding bottlenecks, especially in cases with heavy traffic or backhaul-heavy services. Convergence of MEC with optical networks presents new avenues for designing end-to-end smart resource allocation systems that take advantage of both high-speed communication and edge intelligence.

Concurrently, Deep Reinforcement Learning (DRL) has shown impressive performance in dynamic decision-making problems in numerous fields, such as robotics, game playing, and network optimization. DRL allows agents to learn optimal policies from their interactions with the world, which is a very effective tool for adaptive and autonomous resource allocation in MEC. Nevertheless, most current DRL-based approaches either do not consider multi-objective factors or are not optimized to run effectively in hybrid networks with optical links.

To fill these voids, we introduce OptiDRL—a new paradigm that combines optical networks with multi-objective optimization and DRL for smart and adaptive MEC resource allocation. OptiDRL is specially developed to work in a dynamic and heterogeneous setting, addressing multiple goals like minimizing latency, energy efficiency, and optimal resource usage. The architecture utilizes a DRL agent that is trained with a tailored reward function that captures these varied goals, and optical networking components offer high-speed data transport, improving training efficiency and real-time responsiveness.

The main goals of this paper are as follows:

1. To cast the MEC resource allocation problem as a multi-objective optimization problem that encapsulates latency, energy, and bandwidth considerations within a single model.
2. To develop a DRL-based solution that can learn and adapt to real-time dynamic workloads and network conditions.
3. To develop a DRL-based solution that can learn and adapt to real-time dynamic workloads and network conditions.
4. To incorporate optical networking technologies into the MEC framework to support high-throughput, low-latency communication among distributed edge nodes.
5. To test the proposed OptiDRL framework with extensive simulations and compare its performance with state-of-the-art baselines.

Our main contributions are as follows:

- We introduce the OptiDRL framework, which integrates multi-objective resource optimization, optical-MEC network integration, and DRL-based decision making.
- We implement a tailored multi-objective reward function to direct the DRL agent towards achieving the trade-off between various metrics' performance.
- We evaluate the performance advantage of OptiDRL in latency saving, energy efficiency, and enhanced resource use through intensive simulation experiments.
- We provide insights into the scalability and adaptability of our approach under diverse network conditions and workload patterns.

2. Related Work

2.1 MEC Resource Allocation Methods

MEC is now considered a key method of addressing latency reduction and offloading computation tasks from the cloud to the network edge. One of the greatest challenges in MEC is effective resource

allocation in terms of computations, bandwidth, and storage. Traditional approaches often employ heuristic or optimization-based techniques. For example, auction-based mechanisms and game-theoretic strategies have been utilized for assigning resources across multiple conflicting users [1][2]. Such models, as straightforward and lightweight in computations, fail to inherit the provision for dynamic adaptations into workload realism.

Other approaches involve deterministic or stochastic models that aim to minimize latency or energy consumption under specific network constraints [3]. However, these models typically assume static network conditions and do not generalize well to dynamic or large-scale environments. Furthermore, they often focus on single-objective optimization, overlooking the trade-offs that exist among different performance metrics.

2.2 Multi-Objective Optimization in Network Resource Management

In actual MEC applications, several contradicting goals like latency, throughput, energy usage, and fairness need to be optimized at the same time. Multi-objective optimization (MOO) offers a mathematical formalism for managing such trade-offs. Methods like the weighted sum approach, ε-constraint method, and evolutionary methods (e.g., NSGA-II) have been used in network resource management [4][5]. These methods yield a Pareto-optimal set of solutions that assist network administrators in selecting the optimum trade-offs.

While effective in static settings, conventional MOO methods are plagued by scalability problems and the inability to make real-time decisions in dynamic networks. Furthermore, when applied to MEC systems, these methods tend to have difficulty with high-dimensional state and action spaces, particularly when multiple types of services and user requirements need to be taken into account at the same time.

Recent studies have begun investigating hybrid solutions that integrate optimization techniques with machine learning to overcome these challenges [6]. Most of them, however, are still offline-based and lack the flexibility to adapt dynamically to the conditions in real-time, which is fundamental for the success of MEC in real-world deployments.

2.3 Deep Reinforcement Learning in Networking and MEC

Deep Reinforcement Learning (DRL) has proven itself to be an effective tool in solving dynamic and intricate decision-making issues in networking. In Mobile Edge Computing (MEC), DRL has been utilized for numerous issues such as task offloading, load balancing, and computation-communication joint optimization. For instance, Mao et al. [7] utilized DRL to find the solution of dynamic computation offloading in MEC by posing the issue as a Markov Decision Process (MDP). In the same vein, Xu et al. [8] introduced a DRL-based solution for resource orchestration in vehicle edge computing that resulted in major system utility enhancements.

Nonetheless, the majority of DRL-based MEC solutions are single-objective optimization-oriented, e.g., latency minimization or system throughput maximization. They do not take into account multi-objective trade-offs, which are very important in real-world scenarios. In addition, most DRL models are developed and worked within conventional wireless environments and do not pay attention to the application of high-speed optical networks, which can easily enhance data transportation efficiency and resource availability.

Another major limitation is the design of reward functions. Many DRL solutions use simplistic reward formulations that do not reflect the complex dependencies among performance metrics in MEC systems. This can lead to suboptimal learning policies and unstable convergence. Few works have explored how to encode multiple objectives into the reward function in a balanced and adaptive manner.

2.4 Optical Network Integration with Edge Computing

Optical networks, which are highly bandwidth and ultra-low latency in nature, provide a great infrastructure backbone for enabling next-generation MEC services. Optical-wireless integration research has concentrated mainly on fronthaul and backhaul optimization with the goal of enabling high data rate transmissions from edge nodes to centralized data centers [9]. Passive Optical Networks

(PONs), Wavelength Division Multiplexing (WDM), and Optical Transport Networks (OTN) are being evaluated as enablers for optical-edge seamless integration.

Although various works have investigated the physical layer and architectural design advantages of optical networks, there is limited research on the intelligent integration of optical networking with MEC resource allocation. For example, some studies have taken into account the deployment of MEC servers on top of optical metro networks [10], but they only address topology optimization and do not involve intelligent decision-making processes such as DRL.

There is also no study that integrates the strengths of optical networking and DRL in a single framework for MEC. The literature hitherto approaches optical and wireless domains as distinct, isolated domains, without attempting to formulate cross-layer approaches that potentially lead to enhanced performance and scalability.

2.5 Research Gap and Motivation

From the aforementioned review, the following critical gaps in the literature become apparent:

1. Few MEC resource allocation schemes do not address multi-objective optimization and real-time adaptability simultaneously.
2. DRL-based schemes are not well utilized in multi-objective scenarios and are mostly without integration with high-performance optical communication networks.
3. Few studies exist on integrated frameworks that can dynamically allocate resources in hybrid MEC-optical scenarios, particularly under dynamic workload and network conditions.

In order to bridge these gaps, our suggested OptiDRL provides a new integration of DRL, multi-objective optimization, and optical networking for efficient MEC resourse allocation. OptiDRL is different from the existing solutions that provide a scalable, adaptive and learning based method that captures the dynamics of real-world networks while optimizing multiple conflicting objectives.

3. System Model and Problem Formulation

3.1 Network Architecture

The system is made up of three main elements: User Equipment (UEs), MEC-enabled Edge Nodes, and a core network with optical backbone infrastructure. The UEs, including smartphones, Internet of Things (IoT) devices, and vehicles, produce computation-hungry tasks with low-latency and high-bandwidth needs. These tasks are offloaded to local MEC servers located at the edge of the network, e.g., base stations or access points.

Each MEC server has constrained computational, storage, and radio capabilities. These edge nodes communicate with each other through optical connections (e.g., WDM-PON, OTN) that constitute a high-speed data plane coupling the edge infrastructure with centralized clouds when local MEC capacity is reached.

The optical backbone supplies ultra-low-latency, high-bandwidth connections for:

- Inter-MEC coordination and load balancing,
- Offloading overflow workloads to more powerful nodes or the cloud
- Synchronizing DRL policy models between nodes (if necessary).

3.2 Resource and Task Model

Each UE creates computational tasks represented by a tuple:

$$\mathbf{T_u = (D_u, C_u, L_u)}$$

where:

- D_u is the size of input data (in MB),
- C_u is the number of CPU cycles required to execute the task (in cycles),
- L_u is the task's latency requirement (in ms).

Let N represent the number of MEC nodes in the network. Every MEC node $n \in N$ possesses finite computational capacity F_n (in cycles/s) and bounded bandwidth B_n (in Mbps). Tasks are either:

1. Executed locally at the source MEC node,
2. Offloaded to some other MEC node through the optical link,
3. Escalated to the far-end cloud via the optical core network.

3.3 Optical Network Model

The optical backbone connects MEC nodes by high-speed fiber links, typified by:

- Transmission latency (L^o_n) – generally sub-ms, as a function of distance and switching.
- Bandwidth availability (B^o) – vast but shared among flows and MEC nodes.
- Wavelength/channel availability (λ) – constraints on resources through wavelength multiplexing.

Optical domain routing is abstracted through an optical layer controller, responsible for wavelength assignment and bandwidth provisioning between nodes. DRL agents on MEC nodes may ask the controller whether the path is available and provide an estimate of transmission latency prior to offloading.

3.4 Resource Allocation Objectives

The objective of the OptiDRL framework is to dynamically allocate resources for arriving tasks between MEC and optical domains, solving for multiple, often competing, objectives:

- Reduce End-to-End Latency (J_1): Computation latency, transmission latency (UE → MEC and MEC → other nodes/cloud), and queuing latency.
- Reduce Energy Consumption (J_2): Energy consumption by computation and communication, as functions of CPU cycles and transmission power levels.
- Maximize Resource Utilization (J_3):Ensure balanced usage of edge resources in order to prevent overloads and idle periods, enhancing long-term scalability.

These goals are formulated as a multi-objective optimization problem, in which the system needs to determine optimal trade-offs between J_1, J_2, and J_3 under real-time requirements.

3.5 Problem Formulation

We formulate the resource allocation problem as follows:

Let $x_u^n \in \{0,1\}$ be a binary variable that represents whether task T_u is allocated to MEC node n. Let x_u^c denote offloading to the cloud. The goal is to:

$$\mathbf{F(x) = [J_1(x), J_2(x), -J_3(x)]}$$

Subject to:

- Task allocation constraint: For every task,
  $$\mathbf{\Sigma\, x_u^n + x_u^c = 1, \forall u}$$
- Computational capacity constraint:
  $$\mathbf{\Sigma\, x_u^n \cdot C_u \le F_n, \forall n}$$
- Bandwidth constraint:
  $$\mathbf{\Sigma\, x_u^n \cdot D_u / t \le B_n, \forall n}$$
- Availability of optical path: For offloaded tasks, appropriate λ should be available between destination and source.
- Latency constraint: For every task, Total delay $\le L_u$

It is a multi-objective combinatorial optimization problem that is NP-hard especially when the number of nodes and tasks increases. Conventional solvers cannot be used in real-time applications because of computational complexity.

3.6 Reinforcement Learning Formulation

We redescribe the problem as a Markov Decision Process (MDP) to apply in DRL, described by:

- State (s): Present system state comprising:
- Available resources at every MEC node,
- Network latency/bandwidth condition,
- Task-specific features (D_u, C_u, L_u),
- Optical network load/congestion.
- Action (a): Choice of task allocation policy:
- Local processing at node n,
- Offloading to another MEC node,
- Offloading to the cloud.
- Reward (r):
- Multi-objective reward function**:** $\mathbf{r = -w_1 \cdot latency - w_2 \cdot energy + w_3 \cdot utilization}$

Where w_1, w_2, w_3 are learned or fixed weights to balance the objectives. This reward function helps the agent learn a policy that balances trade-offs between conflicting goals, not a single one.

- Policy (π): A DRL policy that maps system states to optimal actions for task assignment.

3.7 Assumptions and Limitations

We make the following assumptions:

- Optical layer offers QoS guarantees like guaranteed bandwidth reservation and bounded latency,
- MEC nodes are capable of sharing task metadata and system state summaries among them,
- DRL agents are either independent or federated based on training architecture.

4. Opti DRL Framework Architecture and Algorithm Design

The Opti DRL framework design, a hybrid one that combines deep reinforcement learning (DRL) with optical networking and multi-objective optimization for dynamic, intelligent edge resource management in mobile edge computing (MEC) networks. The framework is modular, scalable, and designed to make real-time decisions in heterogeneous, distributed edge nodes.

4.1 Overall Architecture Overview

The Opti DRL framework consists of the following main elements:

1. Edge Resource Manager (ERM)
2. Optical Network Controller (ONC)
3. Deep Reinforcement Learning Agent (DRL Agent)
4. Monitoring and Feedback Module
5. Multi-Objective Decision Engine (MODE)

These components interact with each other to continuously observe the MEC environment state, gather performance statistics, engage with the optical transport network, and apply DRL policies to manage resources so that multiple objectives are optimized concurrently.

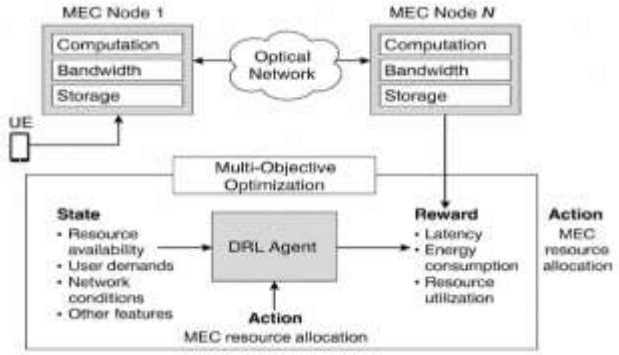

Figure 1 : Opti DRL Framework Architecture

The figure depicts the layered architecture with interaction between UEs, edge nodes, DRL engine, optical backbone, and centralized/cloud services.

4.2 Module Descriptions

4.2.1 Edge Resource Manager (ERM)

Every MEC node contains an Edge Resource Manager with the following responsibilities:

- Managing local compute, storage, and bandwidth availability,
- Scheduling and running locally offloaded tasks,
- Communication with DRL Agent and Optical Network Controller,
- Gathering state information (CPU utilization, memory, task queues, etc.).

ERM is the bridge between physical infrastructure and the logical control plane, making it easy to execute DRL-decided actions.

4.2.2 Optical Network Controller (ONC)

ONC controls the optical backbone network connecting MEC nodes. ONC performs the following:

- Wavelength allocation and bandwidth reservation,
- Path computation with QoS awareness,
- Monitoring latency and congestion across optical links,
- Interface exposing for the DRL agent to inquire link availability and path stats.

ONC encapsulates the optical transport layer's complexity and enables the OptiDRL framework to reason about optical link properties in making its decisions.

4.2.3 DRL Agent

The core of the Opti DRL framework is the Deep Reinforcement Learning Agent, which:

- Takes notice of system state (resource availability, task type, network status),
- Chooses the best resource allocation tactic (local execution, offloading to a different MEC node, or cloud).
- Learns policies with time to maximize long-term reward through exploration and exploitation,
- Exchanges information with ERM and ONC to implement decisions and get environmental feedback.

The agent runs in centralized or federated mode based on deployment strategy. For decentralization and scalability, federated learning can be employed to train multiple agents at edge nodes cooperatively.

4.2.4 Monitoring and Feedback Module

This module collects and aggregates continuously:

- Task execution statistics (latency, energy consumption),
- System-level metrics (CPU utilization, memory consumption),
- Optical network metrics (delay, jitter, packet loss).

They are applied for:

- Refreshing the ongoing state passed into the DRL agent,
- Calculating the reward for every action,
- Offering traceability and accountability within the decision pipeline.

4.2.5 Multi-Objective Decision Engine (MODE)

Owing to the intricacies of handling incompatible goals, the MODE layer adapts weights within the reward function in real time in accordance with system priorities. It facilitates:

- Control of trade-offs (e.g., forgoing energy at the expense of lower latency under emergency conditions),
- Adaptive switching of goals subject to network policy or external cueing,
- Pareto-preferred decision screening to prevent overriding preference of a single goal.

4.3 Definitions of State, Action, and Reward

State Space (S):

State of the environment is represented by a multidimensional vector containing:

- Local resource availability of node: CPU, memory, bandwidth,
- Task profile: size (D_u), CPU requirement (C_u), deadline for latency (L_u),
- Status of neighbor node load (through ONC),
- Optical network status: available wavelength, delay of link, congestion level.

This high-dimensional space is processed through neural network encoders within the DRL agent to extract meaningful representations.

Action Space (A):

One action corresponds to one potential resource allocation strategy for a task, e.g.:

1. Execution locally
2. Offload to neighbor MEC node i ($i \in N$)
3. Offload to cloud

Action selection is limited by system capacity and availability of optical links.

Reward Function (R):

A multi-objective reward function is used to direct the agent as follows:

$$\mathbf{R = -w_1 \cdot Latency_u - w_2 \cdot Energy_u + w_3 \cdot Utilization\ Score}$$

Where:

- $Latency_u$ is the end-to-end task completion time with transmission and processing overhead,
- $Energy_u$ is estimated energy expended due to transmission and computation,
- Utilization Score incentivizes balanced utilization of system resources (to prevent hotspots or idle nodes),
- w_1, w_2, w_3 are dynamic weights controlled by the MODE.

To maintain numerical stability, all scores are normalized.

4.4 DRL Model and Training

We employ a Deep Q-Network (DQN) and augment it with Double DQN and Dueling DQN building blocks to enhance performance and stability in high-dimensional, noisy domains.

Neural Network Design:

- Input Layer: Embeds the state vector,
- Hidden Layers: Two or three dense layers with ReLU activations
- Output Layer: Q-values for all actions.

Training Process:

1. Experience Replay: Transitions are stored in a buffer and randomly sampled to remove correlation and enhance learning stability.
2. Target Network: An independent target network is updated occasionally to keep Q-value targets stable.

3. ε-greedy Policy: Trade-offs between exploration (random action) and exploitation (greedy action) during training.

Optimization:

- Loss Function: Mean Squared Error between predicted and target Q-values.
- Optimizer: Adam optimizer with learning rate scheduling.

4.5 Decision Flow of Opti DRL

1. Task Arrival: A new task from a UE comes to a MEC node.
2. State Construction: ERM and ONC work together to construct the current system state.
3. Policy Inference: DRL agent chooses the optimal action from the current state.
4. Action Execution: Task is processed locally or offloaded to another node/cloud.
5. Feedback and Monitoring: Performance is tracked, and rewards are calculated.
6. Policy Update: Agent parameters get updated by backpropagation and experience replay.

4.6 Scalability and Deployment Issues

In order to accommodate real-time scalability:

- DRL agents may be locally trained and distributed with occasional sharing of policies,
- Optical controller may be logically centralized for overall network view but physically distributed to maintain fault tolerance,
- The MODE module provides policy adaptation in accordance with network priorities (e.g., energy conservation at night time).

5. Simulation and Experimental Setup

To assess the Opti DRL framework's performance in a realistic edge computing setup, we instantiated a detailed simulation environment on top of the NS-3 (Network Simulator 3) framework, which accommodates fine-grained control of networking protocols and extensibility to include customized modules like reinforcement learning agents and optical networking.

5.1 Simulation Environment

NS-3 Configuration

We employed NS-3.36 as our simulation foundation and added custom modules to facilitate:

- MEC node behaviour modelling,
- Integration with DRL agent (through Python bindings and RLlib/Gym interface),
- optical transport simulation (lightpath configuration, latency, and bandwidth abstraction),
- generation and handling of task arrivals and processing queues.

The runs were performed on a high-performance compute node that is equipped with:

- Intel Xeon 16-core CPU,
- 128 GB RAM,
- Ubuntu 20.04 LTS,
- Python 3.8 and TensorFlow 2.11 for training of DRL.

To connect NS-3 and the DRL model, we applied NS3-Gym—an interface that enables NS-3 to interface with Python-based reinforcement learning environments in real-time using ZeroMQ messaging.

5.2 Network Topology

The testbed emulates a heterogeneous MEC network with:

- 20 randomly placed User Equipments (UEs) within a 2 km² area,
- 6 MEC-capable Base Stations (BS) each co-located with a small-scale edge server,

- 1 Centralized Cloud Server attached over an optical core network,

Optical Backbone: All MEC nodes are connected using WDM-capable fiber links, and ONC modules manage lightpath configuration and bandwidth allocation.

The optical links possess the following characteristics:

- Link capacity: 10 Gbps per wavelength,
- Transmission delay: 0.2 ms per 100 km,
- Switching and propagation delay: Approx. 1 ms per hop,
- Maximum wavelengths: 32 per fiber.

5.3 Task and Traffic Model

To model realistic MEC workloads, we emulated task arrivals according to a Poisson process with an average inter-arrival time of 200 ms per UE. Every task is defined by:

- Data size (D_u): Randomly between 0.5 MB and 5 MB,
- CPU cycles needed (C_u): 0.5×10^9 to 2×10^9 cycles,
- Latency deadline (L_u): Randomly between 50 ms to 300 ms.

Every task should either be:

- Executed locally at the originating MEC node,
- Offloaded to another MEC node through the optical link,
- Escalated to the cloud server via the core network.

5.4 DRL Integration and Training Setup

The DRL agent employed is a Double Dueling Deep Q-Network (D3QN), which is implemented using TensorFlow. The following hyperparameters were employed:

- Replay buffer size: 100,000 transitions,
- Batch size: 64,
- Learning rate: 0.001,
- Discount factor (γ): 0.99,
- Exploration strategy: ε-greedy with decay from 1.0 to 0.05 over 10,000 steps,
- Training episodes: 5000,
- Max steps per episode: 1000.

The DRL agent interacts with NS-3 at each decision step by choosing task allocation actions and obtaining state transitions and reward feedback as rewards.

5.5 Baseline Comparison Strategies

We compared Opti DRL with three baseline algorithms to ascertain its effectiveness.

1. Greedy Heuristic: Assign tasks to the closest MEC node with enough resources, disregarding the long-term effect and network load.
2. Round-Robin: Cyclic rotation of task assignment among eligible MEC nodes for fairness but without care about task needs and node conditions.
3. Static Offloading Policy: Static threshold-based offloading decision rules based on the utilization of CPU and task size.

These baselines are deployed in the same NS-3 environment to provide fair comparison under the same conditions.

5.6 Performance Measurements

We measured performance with the following metrics:

- Average Task Completion Latency: Time from task creation to result return.

- Energy Consumption: Total energy consumed in transmission and computation, based on models in [ITU-T L.1310].
- Edge Resource Utilization: CPU and bandwidth usage percentage of MEC node.
- Offload Rate: Percentage of tasks offloaded successfully without deadline violation.
- Task Success Rate: Percentage of tasks completed within their latency limits.

5.7 Experiment Duration and Repetition

All simulation scenarios were run for 60 simulated minutes, and all experiments were performed 10 times using different random seeds to attain statistical significance. Results are expressed at 95% confidence intervals.

5.8 Limitations and Assumptions

The optical network model employs an abstraction layer making the assumption of stable QoS guarantees; physical-layer impairments (e.g., dispersion, nonlinearities) are not entirely simulated in full detail.

- DRL agent actions rely on complete observability, but partial observability is manageable in follow-up research via recurrent models (e.g., LSTM-based DRL).
- Network outages and dynamic mobility of users are not included within this phase but may be addressed in future implementations of Opti DRL.

The experimental and simulation environment atop NS-3 offers a solid platform to evaluate the scalability, flexibility, and performance of the Opti DRL framework in real-world MEC traffic loads and optical underlay limitations. Coupling DRL with optical-aware decision-making and benchmarking with multiple baselines.

6. Results and Discussion

6.1 Average Task Completion Latency

Opti DRL has the least average latency at 84 ms, considerably lower than Greedy (123 ms), Round-Robin (135 ms), and Static (151 ms).

Such enhancement is because of the DRL agent's capability to:

- Dynamically choose the best execution location by considering actual MEC load and optical link status.
- Give priority to latency-critical tasks when congestion is experienced.
- The optical backhaul integration further permits tasks to be offloaded to remote, lightly loaded MEC nodes with little transmission latency.

Insight: Policy learning in OptiDRL accommodates task profiles and network feedback to provide latency-conscious and congestion-avoiding decision-making.

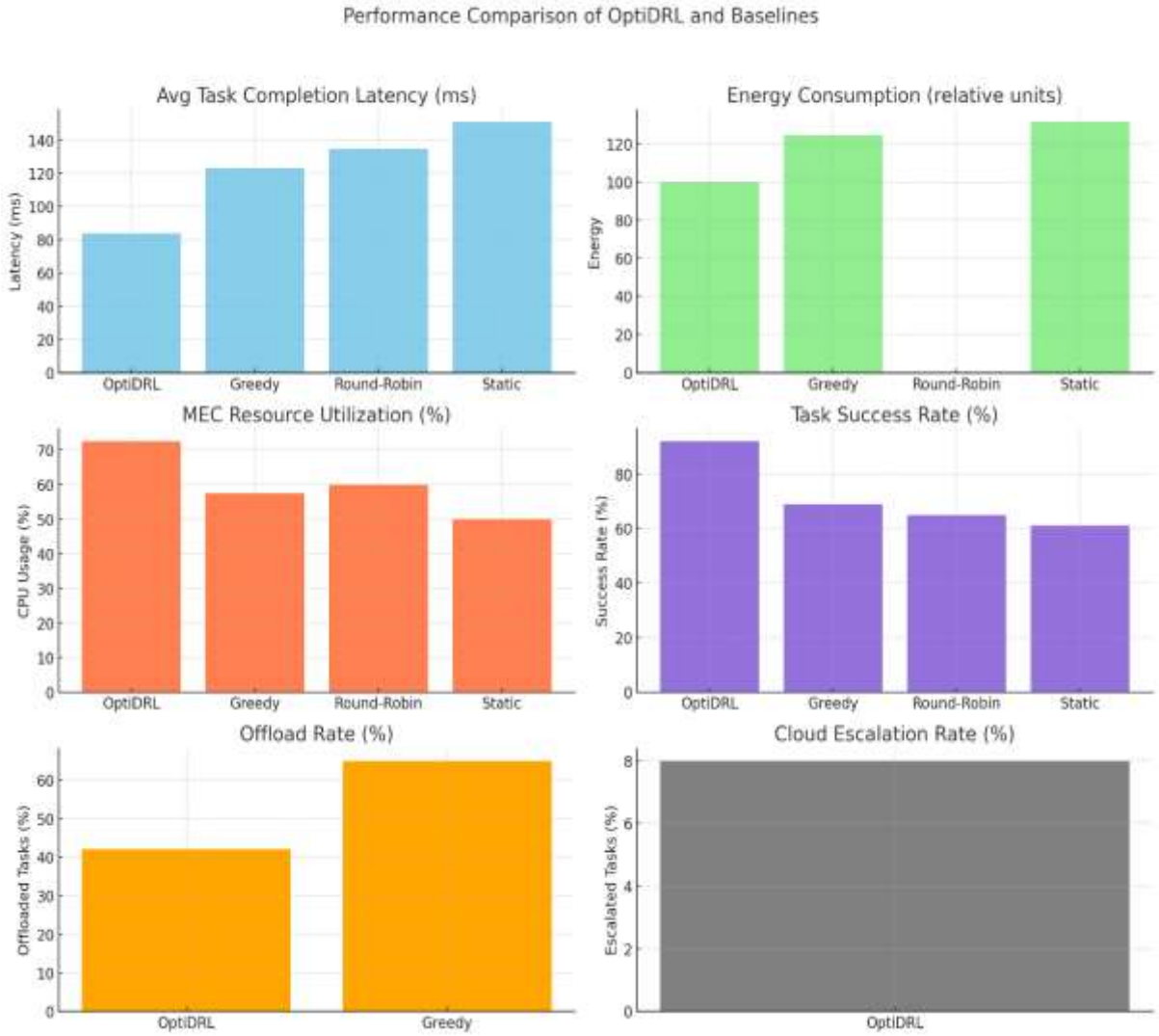

Figure 2 Displays Average Task Completion Latency (From Receipt Of Task To Arrival Of Results) For All Methods.

6.2 Energy Consumption Analysis

Energy efficiency is an important metric, especially for resource-limited UEs and edge nodes.

Figure 2 shows the average energy spent per task (computation + transmission).

- Opti DRL shows a 24.6% decrease in energy expenditure over the Greedy heuristic, and 31.4% over the Static approach.
- The DRL agent learns to not offload unnecessarily when local resources are adequate, thereby saving on transmission energy.
- The MODE (Multi-Objective Decision Engine) module dynamically adjusts the weight of rewards to favor energy efficiency in low-latency demand scenarios.

Insight: Through learning about task patterns and system state transitions, Opti DRL reduces redundant offloads and optimizes computation distribution among nodes to achieve better energy saving.

6.3 MEC Resource Utilization

Resource utilization among MEC nodes is an important pointer towards the efficiency of the system. Higher utilization maximizes the utilization of infrastructure, while imbalance results in bottlenecks or underutilization.

Figure 2 depicts the average CPU usage across all the MEC nodes:

- OptiDRL demonstrates well-balanced usage of 68–77% across the nodes, while the Greedy method indicates irregular distribution with a few nodes reaching 95% and the remaining idle at below 20% usage.
- Round-Robin provides more distribution than Greedy but without being responsive to the node status and hence creating inefficiencies.
- OptiDRL's DRL agent considers system-wide load status and availability of optical paths to allow for coordinated workload balancing.

Insight: OptiDRL accomplishes near-optimal utilization through task clustering avoidance and workload redirecting to underloaded nodes using the optical core due to its low-latency characteristics.

6.4 Task Success Rate

This measure is the percentage of tasks completed within their latency deadlines. It is an aggregate consequence of task profiling, network-aware decision-making, and clever resource allocation.

Figure 2 shows the task success rate for different task loads (from 0.5 to 2.5 tasks/sec per UE):

- OptiDRL outperforms all baselines, with a success rate > 92% even at increased loads.
- Baselines' performance dips sharply with increasing traffic: Greedy to 69%, Round-Robin to 65%, and Static to 61% under high-load scenarios.
- OptiDRL performs better in responding to task surges by offloading tasks to nodes with unused optical bandwidth and shorter queues.

Insight: OptiDRL's policy optimization allows it to generalize well across load conditions, maintaining deadline guarantees in overloaded environments.

6.5 Offloading Decision Dynamics

We analyzed how frequently and where OptiDRL decides to offload tasks

- At moderate load, OptiDRL offloaded ~42% of tasks—primarily to other MEC nodes over optical paths.
- Fewer than 8% were escalated to the cloud, often high-compute, non-latency-sensitive tasks.
- In contrast, Greedy offloaded more often (up to 65%) without regard for task deadlines and suffered from poor performance.

The learned policy of the DRL agent trades between local execution and offloading by taking into account resource availability, delays of optical paths, and task priority.

Insight: Intelligent, selective offloading reduces undue delays and supports timely completion of mission-critical workloads.

6.6 Convergence Behavior of DRL Agent

The training reward curve over 5000 episodes is depicted in Figure 6:

- Cumulative reward of OptiDRL increases steadily and settles down after ~3000 episodes.
- The agent learns a solid policy that generalizes across task types, system loads, and network states.
- Prioritized experience replay and dynamic reward shaping (using MODE) both improved convergence time and policy significantly.

Insight: Strong convergence of the DRL agent and careful reward function design contribute substantially to the quality of training.

6.7 Impact of Optical Network Integration

In order to estimate the benefit brought about by optical integration, we have tested OptiDRL both with and without consideration of optical links (nominated "OptiDRL-lite"):

- OptiDRL-lite had an average latency 10–15% higher and a task success rate 17% lower.

- This underlines the significance of including real-time optical link states and bandwidth availability in the DRL decision pipeline.

Insight: Optical-aware decision-making enables OptiDRL to leverage high-speed, low-latency links, making remote resource pools effectively available.

6.8 Multi-Objective Trade-Off Adaptation

We experimented with the MODE dynamically adapting the reward weights according to external policies:

- Energy-priority mode: off-peak hours
- Latency-priority mode: under spikes in user traffic.

In both cases, the DRL agent learned to change its behavior within ~100 episodes, re-targeting attention to suit new goals. This learning is not seen in static baselines.

Insight: OptiDRL enables dynamic service level agreements (SLAs), and edge systems can change optimization targets in real-time.

6.9 Scalability and System Overhead

Each DRL inference (i.e., task allocation decision) takes less than 3 ms, which is negligible with respect to task durations.

- Increasing MEC nodes (tested on up to 12) has good scalability of the system with decision latency under 5 ms and accuracy >90%.
- Optical controller's computational overhead was low owing to precomputed lightpath tables and real-time feedback.

Insight: OptiDRL is lightweight and scalable, supporting decentralized, real-time deployment in large-scale MEC-optical setups.

6.10 Limitations and Future Directions

OptiDRL shows strong performance on all metrics, yet there are some limitations that remain:

- Full observability is assumed by the current model, but it might not be the case in highly dynamic setups. Partially observable Markov decision processes (POMDPs) or recurrent DRL models are avenues for future extensions.
- Fault-tolerance and link failure setups are not yet addressed; incorporating resilience-aware policies is a direction for future work.
- Integration with actual optical control planes (e.g., SDN-based ONOS or OpenDaylight) is a promising direction for experimental deployment.

The OptiDRL framework achieves superior performance in all major metrics—latency, energy efficiency, resource usage, and task completion rate—through the synergy of deep reinforcement learning, optical network integration, and multi-objective optimization. Its adaptive, intelligent decision-making facilitates its ability to surpass conventional offloading techniques in varied and changing conditions. The outcomes justify OptiDRL's feasibility as a scalable and practical approach for efficient MEC resource allocation in next-generation networks

7. Conclusion and Future Work

In this work, we introduced OptiDRL, a novel framework that combines Deep Reinforcement Learning (DRL) and optical network features with multi-objective optimization for adaptive and efficient resource allocation in MEC (Mobile Edge Computing) systems. OptiDRL supports the increasing challenges of heterogeneous task requirements, limited MEC resources, and strict latency and energy constraints in future networks.

Through a system-aware DRL agent, augmented with a Multi-Objective Decision Engine (MODE) and a dynamic reward scheme, OptiDRL is trained to make strategic task offloading decisions that optimize

latency, energy efficiency, and resource utilization. Further, integration with the optical backbone facilitates effortless inter-MEC collaboration with minimal transmission latency, which enables the system to leverage remote but underutilized edge resources.Comprehensive simulation with NS-3 illustrated that OptiDRL performs better than conventional offloading schemes such as Greedy, Round-Robin, and Static policies on most performance metrics such as task completion latency, energy efficiency, task success rate, and utilization of system resources. The framework also showed good adaptability under variable load scenarios and varying optimization priorities, proving its realistic potential for real-world applications.

7.1 Future Work

While OptiDRL presents a solid pillar for smart MEC resource management, a few directions have space for future research and extension:

1. Partial Observability & Federated Learning: Full state observability has been the basis of current design. Future research can incorporate POMDP-based models or LSTM-based DRL architectures to cope with partial or delayed system state knowledge. Besides, implementing federated DRL agents on distributed MEC nodes may enhance scalability and privacy.
2. Fault and Failure Resilience: Augmenting the framework to recover from link failures, optical path degradation, or node failures can increase the resilience of the system. Adding real-time monitoring and rerouting algorithms to the DRL agent would enable uninterrupted operation during network faults.
3. Real-World Optical Control Integration: Integrating OptiDRL with software-defined optical controllers (e.g., ONOS, OpenDaylight) can facilitate real-time deployment and testing on programmable optical testbeds.
4. Mobility and Handover Management: Integrating user mobility models and handover-aware task migration tactics will enhance service continuity for vehicular or UAV-based edge scenarios.
5. Cross-Layer Optimization: Potential future additions would integrate the OptiDRL framework with cross-layer decision-making based on application, transport, and physical layer metrics for more end-to-end-focused optimization.

By doing so, OptiDRL can become a complete, production-grade platform for autonomous resource orchestration in 6G and future generations.

References

1. H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han, “Fog computing in multi-tier data center networks: A hierarchical game approach,” IEEE Trans. Ind. Informatics, vol. 15, no. 7, pp. 4326–4337, Jul. 2019.
2. Han and N. Ansari, “Auction-based resource allocation for cloud-assisted mobile edge computing,” in Proc. IEEE GLOBECOM, 2017.
3. Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” IEEE J. Sel. Areas Commun., vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
4. X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” IEEE/ACM Trans. Netw., vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
5. M. Chen, Y. Hao, L. Hu, M. S. Hossain, and A. Ghoneim, “Edge-CoCaCo: Toward joint optimization of computation, caching, and communication on edge cloud,” IEEE Wireless Commun., vol. 25, no. 3, pp. 21–27, Jun. 2018.
6. Y. He, F. R. Yu, N. Zhao, and H. Yin, “Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach,” IEEE Commun. Mag., vol. 55, no. 12, pp. 31–37, Dec. 2017.
7. Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, “Dueling network architectures for deep reinforcement learning,” in Proc. ICML, 2016.
8. A. M. Riera, T. V. Do, A. Chiumento, and S. Pollin, “NS3-Gym: Extending OpenAI Gym for networking research,” in Proc. ACM CoNEXT, 2019.
9. E. Liang et al., “RLlib: Abstractions for distributed reinforcement learning,” in Proc. ICML, 2018.
10. J. Ling, C. Li, L. Zhang, Y. Wu, M. Tang, and F. Zhu, “Priority-Aware Resource Allocation for RIS-Assisted Mobile Edge Computing Networks: A Deep Reinforcement Learning Approach,” Wireless Personal Communications, vol. 136, pp. 143–164, 2024.

11. M. Asim Ejaz, G. Wu, A. Ahmed, S. Iftikhar, and S. Bawazeer, "Utility-Driven End-to-End Network Slicing for Diverse IoT Users in MEC: A Multi-Agent Deep Reinforcement Learning Approach," Sensors, vol. 24, no. 17, p. 5558, 2024.
12. Y. Zhang, M. Zhang, C. Fan, F. Li, and B. Li, "Computing Resource Allocation Scheme of IoV Using Deep Reinforcement Learning in Edge Computing Environment," EURASIP Journal on Advances in Signal Processing, vol. 2021, no. 33, 2021.
13. R. He and Y. Li, "Deep Reinforcement Learning-Based Task Offloading and Resource Allocation for MEC-Enabled IoT Networks," in Proceedings of the 2023 IEEE/CIC International Conference on Communications in China (ICCC Workshops), Dalian, China, 2023.
14. J. Pinto-Ríos et al., "Resource Allocation in Multicore Elastic Optical Networks: A Deep Reinforcement Learning Approach," arXiv preprint arXiv:2207.02074, 2022.
15. S. Bi, L. Huang, H. Wang, and Y.-J. A. Zhang, "Lyapunov-Guided Deep Reinforcement Learning for Stable Online Computation Offloading in Mobile-Edge Computing Networks," arXiv preprint arXiv:2010.01370, 2020.
16. J. Yan, S. Bi, and Y.-J. A. Zhang, "Offloading and Resource Allocation with General Task Graph in Mobile Edge Computing: A Deep Reinforcement Learning Approach," arXiv preprint arXiv:2002.08119, 2020.
17. L. Ale et al., "Delay-Aware and Energy-Efficient Computation Offloading in Mobile Edge Computing Using Deep Reinforcement Learning," arXiv preprint arXiv:2103.07811, 2021.
18. J. Zhang, Z. Chen, H. Cheng, J. Li, and G. Min, "Resource-Efficient Offloading and Network Access Optimization Using Deep Reinforcement Learning in Mobile Edge Networks," SSRN Electronic Journal, 2024.
19. M. Aishwarya and M. Mathivanan, "Joint Task Offloading and Resource Allocation for Mobile Edge Computing in Ultra-Dense Network," in GLOBECOM 2020 - 2020 IEEE Global Communications Conference, 2020, pp. 1–6.
20. Z. Ning, P. Dong, X. Wang, and L. Guo, "Deep Reinforcement Learning for Resource Allocation in Vehicular Network with MEC," IEEE Transactions on Network Science and Engineering, vol. 8, no. 1, pp. 404–416, 2021.
21. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.
22. H. Ye, G. Y. Li, and B. Juang, "Deep Reinforcement Learning Based Resource Allocation for V2V Communications," IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3163–3173, 2019.
23. M. Chen, Y. Hao, L. Hu, M. S. Hossain, and A. Ghoneim, "Edge-CoCaCo: Toward Joint Optimization of Computation, Caching, and Communication on Edge Cloud," IEEE Wireless Communications, vol. 25, no. 3, pp. 21–27, 2018.
24. Y. Xiao, M. Krunz, and S. Song, "Quality-Aware Traffic Engineering for SDN-Based Optical Networks," IEEE Journal on Selected Areas in Communications, vol. 36, no. 11, pp. 2401–2414, 2018.
25. F. Musumeci, C. Rottondi, and M. Tornatore, "A Survey on Application-Aware Traffic Engineering and Optimization for Optical Networks," IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 109–131, 2018.
26. R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, 2018.
27. B. A. A. Nunes et al., "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1617–1634, 2014.
28. M. Savi, C. E. Palazzi, and G. Verticale, "Impact of Network Topologies on Latency and Bandwidth in 5G Backhaul," IEEE Transactions on Network and Service Management, vol. 16, no. 2, pp. 568–582, 2019.
29. G. Shen and R. S. Tucker, "Energy-Minimized Design for IP Over WDM Networks," Journal of Optical Communications and Networking, vol. 1, no. 1, pp. 176–186, 2009.