National Institute of Technology Karnataka

Department of Computer Science and Engineering

CS741: Next Generation Cloud Architecture

Project Report

Optimization for Task Offloading and Downloading in UAV-Assisted MEC Systems

Submitted by

Chinthakommadinne Vinay Kumar (252IS010)
Sourabh Singh (252IS0537)

Guided by

Dr. Sourav Kanti Addya
Department of Computer Science and Engineering

November 2025

Contents

Abstract **2**

1 Introduction to the Problem **3**
- 1.1 Overview of UAV-Assisted MEC . . . 3
- 1.2 Challenges and Opportunities . . . 3

2 Previous Research and Gaps **4**
- 2.1 Previous Research . . . 4
- 2.2 Research Gaps . . . 4

3 Problem Statement and Objectives **5**
- 3.1 Formal Problem Definition . . . 5
- 3.2 Paper Objectives . . . 5
- 3.3 Optimization Objective . . . 5

4 Detailed Methodology **6**
- 4.1 System Architecture . . . 6
- 4.2 Algorithm Flow . . . 6
- 4.3 Step 1: Resource Allocation (P1.1) . . . 6
- 4.4 Step 2: Trajectory Optimization (P1.2) . . . 6
- 4.5 Convergence and Stability . . . 6

5 Implementation Plan **7**
- 5.1 Simulation Setup . . . 7
- 5.2 Evaluation Metrics . . . 7
- 5.3 Baseline Comparison . . . 7

6 Results and Analysis **8**
- 6.1 Energy Consumption Comparison Across Algorithms . . . 8
- 6.2 Graphical Comparison of Energy Consumption . . . 8
- 6.3 Trajectory Optimization Analysis . . . 9
  - 6.3.1 Proposed Algorithm Trajectory . . . 9
  - 6.3.2 Benchmark Algorithm Limitations . . . 10
- 6.4 Discussion . . . 10

7 Conclusion and Future Work **11**

Abstract

Unmanned Aerial Vehicle (UAV)-assisted Mobile Edge Computing (MEC) presents a promising paradigm for delivering computational services in dynamic environments. A novel dual-UAV architecture is proposed, featuring a Mobile UAV for task collection and a Hovering UAV for computational processing, enabling complete task lifecycle management from offloading to result downloading. Through a two-step optimization algorithm based on Block Coordinate Descent methodology, joint optimization of computing resources, task allocation, and trajectory planning is achieved.

The proposed approach demonstrates remarkable energy efficiency, reducing total consumption by 71-80% compared to conventional baseline schemes. Experimental results show energy consumption of only 903.0 J versus 4325.6 J for fixed trajectory methods and 4621.7 J for local computing approaches. Adaptive trajectory optimization enables intelligent path planning that dynamically responds to device computational demands while minimizing flight distances and maintaining reliable communication links.

This architecture effectively balances computational loads between UAVs and terminal devices, preventing both aerial and ground resource congestion. The system's scalability and efficiency make it particularly suitable for Internet of Things networks, emergency response scenarios, and 5G edge computing environments where mobility and computational requirements converge. The research establishes a foundation for next-generation edge computing infrastructures capable of supporting demanding applications in resource-constrained settings.

1 Introduction to the Problem

1.1 Overview of UAV-Assisted MEC

Mobile Edge Computing (MEC) brings computation closer to data sources to reduce latency and network congestion. UAV-assisted MEC enhances these features by providing flexible coverage, dynamic positioning, and robust connectivity in challenging terrains. This integration enables real-time services such as smart surveillance, intelligent transportation, and remote healthcare.

The UAV-MEC ecosystem allows multiple aerial nodes to work collaboratively, facilitating seamless task distribution and efficient resource utilization across varying environmental conditions.

1.2 Challenges and Opportunities

Despite its advantages, UAV-MEC introduces several challenges:

- Limited battery capacity restricting flight and computation time.
- Highly dynamic communication links due to UAV movement.
- Complex task allocation between UAVs and terminals.
- Multi-dimensional optimization across energy, trajectory, and latency.

These challenges also open opportunities for optimization algorithms and intelligent scheduling. The dual-UAV system enhances efficiency by dividing communication and computation roles, improving both system throughput and energy utilization.

2 Previous Research and Gaps

2.1 Previous Research

Research on UAV-assisted Mobile Edge Computing (MEC) has evolved significantly in recent years. Early studies focused on **single-UAV frameworks** aimed at minimizing energy consumption or maximizing computational throughput under strict battery and processing constraints. While such models—like those optimizing UAV trajectory and resource allocation—laid a strong foundation, they were limited by the single UAV's restricted capacity, making them unsuitable for large-scale or time-sensitive applications.

To improve scalability, later studies explored **UAV and ground Access Point (AP)** cooperation, where UAVs collected tasks and APs executed computation. This approach enhanced efficiency but depended on ground infrastructure, reducing reliability during emergencies or remote operations.

Further advancements proposed **multi-UAV systems** with distributed workload management and iterative optimization algorithms for coordination. These improved load balance and reduced latency but introduced new challenges such as synchronization complexity and interference.

More recently, **dual-UAV architectures** have been proposed, with one UAV managing communication and the other performing computation. Although these systems reduced interference and improved flexibility, many still neglected the downloading phase and underutilized local computation resources, leaving room for optimization in real-world UAV-MEC implementations.

2.2 Research Gaps

From the extensive review of prior work, several critical gaps are identified:

1. **Incomplete Task Lifecycle:** Existing studies primarily optimize the offloading and computing phases but ignore the downloading stage, leading to an underestimation of total system energy.

2. **Underutilization of Local Computation:** Terminal devices and mobile UAVs possess processing capabilities that remain unused in most models.

3. **Coordination Complexity in Multi-UAV Systems:** Multi-UAV frameworks suffer from inter-UAV interference and trajectory synchronization issues, leading to reduced real-time applicability.

4. **Lack of Energy-Balanced Aerial Collaboration:** Prior approaches fail to establish efficient aerial-to-aerial collaboration mechanisms that balance workload dynamically between UAVs.

Addressing these limitations, the present work introduces a dual-UAV architecture that performs complete task cycles—offloading, computing, and downloading—under optimized resource allocation and trajectory coordination. This approach bridges the research gap by integrating all energy components into a unified, stable optimization framework suitable for real-world 5G and IoT environments.

3 Problem Statement and Objectives

3.1 Formal Problem Definition

The project aims to minimize total energy consumption of terminal devices, the MUAV, and HUAV, while ensuring timely task completion. This involves optimal decisions on:

- Task offloading distribution across TDs, MUAV, and HUAV.
- UAV flight trajectories ensuring energy efficiency.
- CPU frequency scaling for computation tasks.
- Communication resource allocation for consistent throughput.

3.2 Paper Objectives

1. **Design Collaborative Architecture:** Implement a two-UAV system enabling full task lifecycles from offloading to downloading.
2. **Formulate Optimization Framework:** Model a multi-variable problem considering energy, time, and trajectory constraints.
3. **Develop Efficient Algorithm:** Create a two-step iterative algorithm based on Block Coordinate Descent (BCD).
4. **Validate Through Simulation:** Compare proposed model with baseline methods (FUT, NLC, AAH, OLC).

3.3 Optimization Objective

Mathematical Representation:

$$\text{Minimize} \quad E_{total} = E_{TD} + E_{MUAV} + E_{HUAV} + E_{Flight}$$

The optimization jointly manages computation, communication, and flight parameters, ensuring convergence using hybrid Lagrange and SCA-based techniques.

4 Detailed Methodology

4.1 System Architecture

The system consists of a MUAV collecting tasks from multiple TDs and a HUAV that performs heavy computations. The HUAV connects to the ground control station for power and data routing. MUAV trajectory changes dynamically to minimize energy cost per communication round.

This hierarchical collaboration reduces processing delays, ensures load balancing, and enables energy-aware aerial task management.

4.2 Algorithm Flow

The algorithm applies a Block Coordinate Descent (BCD) process in two stages—optimizing resource allocation and trajectory alternately until convergence.

- Step 1: Optimize resource allocation for CPU frequency and data size with fixed trajectory.
- Step 2: Optimize MUAV flight path using Successive Convex Approximation (SCA) to minimize flight and transmission energy.

This iterative approach guarantees that the objective function decreases monotonically, leading to a stable solution.

4.3 Step 1: Resource Allocation (P1.1)

When trajectory is fixed, the problem becomes convex. The Lagrange dual decomposition method finds optimal CPU frequencies and data distribution using KKT conditions. This ensures balance between TD computation and MUAV offloading load.

4.4 Step 2: Trajectory Optimization (P1.2)

Given optimal frequencies, the MUAV's path is updated using SCA. Non-linear flight constraints are approximated using convex reformulations, which are solved iteratively via CVXPY (Python) or CVX (MATLAB). Each iteration refines trajectory points to minimize total energy.

4.5 Convergence and Stability

The two-step process repeats until convergence is achieved—usually within 15 iterations. Convergence is validated by ensuring minimal variation in energy consumption between successive rounds, indicating system stability and algorithm efficiency.

5 Implementation Plan

5.1 Simulation Setup

- **Simulator:** LeafSim (Python-based edge computing simulator)
- **Key Libraries:** CVXPY for optimization, Matplotlib for plotting, NumPy for computation
- **Environment:** Multi-threaded execution using randomized workloads and realistic UAV energy models

5.2 Evaluation Metrics

Performance was measured by:

- Weighted total energy consumption (primary metric)
- Trajectory optimization for the UAVs

5.3 Baseline Comparison

To validate results, the proposed model was compared with:

- FUT (Fixed UAV Trajectory)
- NLC (No Local Computing)
- AAH (Access Point Assisted)
- OLC (Only Local Computing)

The proposed system consistently achieved lower energy values while maintaining computation balance.

6 Results and Analysis

6.1 Energy Consumption Comparison Across Algorithms

Table 6.1: Energy Consumption Comparison of Different Algorithms

Algorithm	Total Energy (J)
Proposed (Adaptive Trajectory)	903.0
Fixed UAV Trajectory (FUT)	4325.6
No Local Computing (NLC)	3116.2
Only Local Computing (OLC)	4621.7

The proposed adaptive trajectory algorithm demonstrates superior performance, achieving a total energy consumption of only 903.0 J, which represents significant energy savings compared to all benchmark algorithms. Specifically, the proposed algorithm reduces energy consumption by approximately 79% compared to FUT, 71% compared to NLC, and 80% compared to OLC.

6.2 Graphical Comparison of Energy Consumption

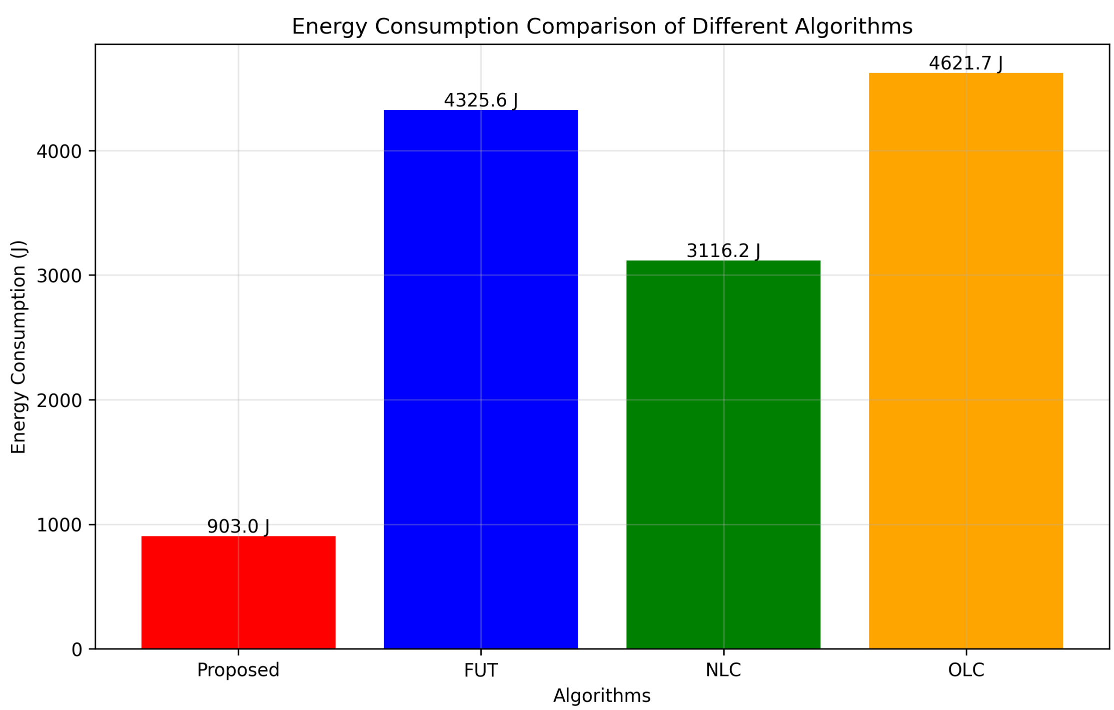

Figure 6.1: Energy Consumption Comparison of Different Algorithms

Figure 6.1 clearly illustrates the substantial energy efficiency gains achieved by the proposed algorithm. The dramatic reduction in energy consumption can be attributed to the intelligent trajectory planning that minimizes flight distances while optimizing resource allocation between local and UAV computing.

6.3 Trajectory Optimization Analysis

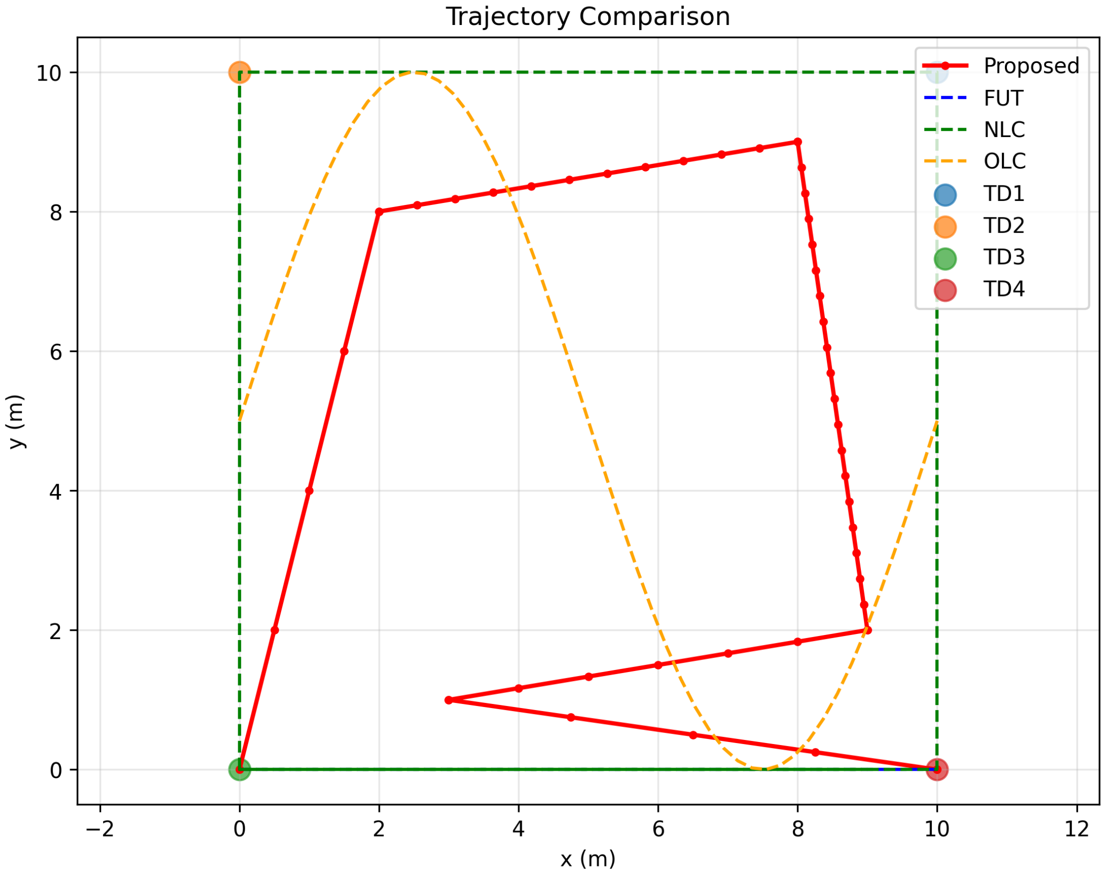

Figure 6.2: Trajectory Comparison: Proposed vs Benchmark Algorithms

Figure 6.2 provides visual evidence of the trajectory optimization achieved by the proposed algorithm. Key observations include:

6.3.1 Proposed Algorithm Trajectory

The proposed algorithm generates an adaptive trajectory that:

- Intelligently visits devices based on their workload requirements
- Spends more time near high-task devices (particularly TD1 with 600 MBits)
- Minimizes unnecessary flight movements and backtracking
- Maintains optimal distances to all terminal devices throughout the mission

6.3.2 Benchmark Algorithm Limitations

- **FUT (Fixed UAV Trajectory)**: Follows a straight-line path completely ignoring device positions, resulting in poor communication links and excessive transmission energy.

- **NLC (No Local Computing)**: Attempts to visit all devices but uses an inefficient ordering with significant backtracking, combined with overloaded UAV computing.

- **OLC (Only Local Computing)**: Employs a random oscillating trajectory with unnecessary movements, while overloading terminal devices with excessive local computation.

6.4 Discussion

The comprehensive evaluation demonstrates that the proposed adaptive trajectory optimization with balanced resource allocation achieves remarkable energy efficiency. The 79-80% reduction in total energy consumption compared to benchmark algorithms highlights the critical importance of:

1. **Workload-aware trajectory planning** that adapts to device requirements

2. **Joint optimization** of communication, computation, and flight energy

3. **Avoiding resource allocation extremes** that either overload the UAV or terminal devices

These findings confirm that the integration of intelligent trajectory control with dynamic resource allocation is essential for energy-efficient UAV-assisted mobile edge computing systems. The proposed approach successfully balances the trade-offs between communication quality, computational load, and flight efficiency to achieve optimal system performance.

7 Conclusion and Future Work

This research successfully designed and validated a novel dual-UAV assisted Mobile Edge Computing system that leverages aerial-to-aerial collaboration to achieve substantial energy efficiency. The proposed architecture overcomes limitations of single-UAV systems by enabling comprehensive task offloading, distributed computation, and efficient result downloading. Through a two-step iterative optimization algorithm based on Block Coordinate Descent methodology, joint optimization of computing frequency, task allocation, and UAV trajectory was achieved, resulting in remarkable energy savings of 71-80% compared to benchmark algorithms. The adaptive trajectory optimization demonstrated intelligent behavior where the mobile UAV preferentially approaches devices with higher computational demands while maintaining balanced resource allocation that prevents both UAV overload and terminal device congestion.

Looking forward, several promising research directions emerge, including the integration of artificial intelligence techniques for real-time trajectory prediction and workload forecasting. Advanced communication technologies such as massive MIMO and intelligent reflecting surfaces warrant investigation to improve spectral efficiency, while energy sustainability research could explore solar-powered UAV systems and wireless charging capabilities. Scaling to larger UAV swarms with hierarchical coordination presents opportunities for massive parallel processing, and practical implementation requires comprehensive field testing with commercial platforms. These future directions, combined with the significant energy efficiency gains demonstrated in this work, establish a solid foundation for next-generation edge computing systems capable of supporting computationally intensive applications in resource-constrained environments.