Trajectory Optimization for Deadline-Constrained UAV-Assisted Mobile Edge Computing Networks

Khalil El Kaaki, Joe Abi Samra

***Abstract*—This paper studies trajectory optimization for UAV-assisted mobile edge computing (MEC) networks with deadline-constrained task offloading. A system model is developed in which IoT devices generate stochastic computation tasks that can be executed locally, offloaded to a terrestrial base station, or served by a mobile UAV-mounted MEC server. Uplink transmission is TDMA-based, and wireless rates depend on distance through an Okumura–Hata path loss model. The objective is to maximize the number of completed tasks within their deadlines by jointly optimizing offloading decisions, TDMA scheduling, and the UAV trajectory. The resulting problem is mixed-integer, nonlinear, and time-varying. Two gradient-based trajectory optimization strategies are proposed: an online model predictive control (MPC) approach and an offline batch optimization approach. Extensive Monte Carlo simulations demonstrate that optimized strategies consistently outperform fixed circular trajectories and no-UAV baselines across task size, network density, and bandwidth sweeps. The gains are most pronounced in communication-limited scenarios where clever positioning mitigates path loss and deadline pressure. The results confirm the beneficial role of trajectory-aware control in enabling reliable, low-latency UAV-assisted MEC.**

I. Introduction

UAVS can serve as airborne base stations (ABSs) capable of delivering both wireless access and computational capability close to where users are in a wireless network. i.e., ABSs can host miniature servers at the network edge, resulting in an ABS-aided multi-edge computing (MEC) network architecture. This configuration provides reliable and fast on-demand service in networks that are congested, or prone to fail, or have limited infrastructure, or all [1]. However, ABS MEC nodes have limited compute resources, and the ABS-user distance imposes rapid path loss growth that degrades channel quality. These factors cap the service quality of an ABS MEC node even when it relieves ground-network load. Therefore, clever ABS trajectory planning is the primary means of making the most of the available resources to compute user tasks in a timely manner. This paper models the network's performance — quantified by the amount of served tasks — as an optimization problem that determines the optimal UAV trajectory subject to task deadlines, minimum quality-of-service (QoS), transmission constraints, and UAV mobility limitations. The formulation jointly captures the coupling between wireless communication, computation latency, and UAV motion, revealing the inherent trade-offs between proximity, coverage, and resource availability.

The remainder of the paper is organized as follows. Section II discusses the established research in the reliability, performance, and potential of ABS-aided MEC networks. Section III presents the system model, including the task generation process, communication model, and computation framework. Section IV formulates the joint trajectory and task completion optimization problem. Section V describes the proposed online and offline optimization methodologies. Section VI reports and discusses the simulation results. Finally, Section VII concludes the paper and outlines directions for future work.

II. Literature Review

UAV-assisted MEC has attracted extensive interest due to its potential in providing computation, connectivity, and sensing capabilities in environments where terrestrial infrastructure is unavailable or insufficient. Existing work spans UAV trajectory design, computation offloading, joint communication–computation optimization, reliability-aware edge services, and machine-learning-based adaptive frameworks. This section reviews the most relevant contributions, including three advanced optimization frameworks that directly relate to our study.

A. UAV-Assisted Wireless Networks and Trajectory Optimization

Early research treats UAVs primarily as aerial base stations whose goal is to maximize coverage or throughput by optimizing altitude, hovering location, or flight trajectory [2]. These models typically rely on probabilistic LoS channel characterization and simplified mobility assumptions. Classical trajectory patterns include static hovering for hotspot coverage, circular paths for periodic service, and deterministic waypoint routes.

While foundational, these approaches do not incorporate computation offloading or MEC capabilities. More importantly, they often assume static user distributions or predictable mobility, limiting their applicability in dynamic IoT and IIoT systems with time-varying workloads.

B. MEC-Enabled UAV Networks

As UAVs evolved to carry computation units, several works explored UAVs as flying MEC servers . These studies investigate binary or partial offloading, CPU frequency control, transmit power scheduling, and basic trajectory adjustments [3]. Joint optimization of communication and computation parameters has shown significant improvements in latency and energy consumption.

However, such contributions frequently assume:

- a single UAV,
- deterministic task arrivals,
- ideal channel knowledge, and
- simplified mobility models.

These assumptions limit real-time applicability, particularly in dense IoT deployments with random traffic and mobility.

C. Joint Communication–Computation–Trajectory Optimization

Several works attempt to jointly optimize UAV trajectory, bandwidth allocation, transmit power, and MEC computation resources . The resulting optimization problems are highly non-convex and are typically addressed using alternating optimization, block coordinate descent, or successive convex approximation [2].

While these frameworks produce strong offline solutions, they require prior knowledge of user positions and task characteristics. As such, they struggle when operating under uncertain, dynamic environments where task arrival patterns, energy levels, and wireless conditions evolve unpredictably over time.

D. Reliability- and QoS-Aware UAV-MEC Systems

Another research direction incorporates reliability guarantees, probabilistic LoS models, outage probability constraints, and latency bounds into MEC optimization . These contributions highlight the importance of robust communication links for mission-critical services. Nonetheless, they generally treat reliability in isolation, without jointly addressing computation deadlines, UAV energy budgets, or dynamic task arrivals.

E. MOALF-UAV-MEC: Adaptive Multi-Objective Optimization Framework

A recent advancement is the MOALF-UAV-MEC framework [3] , which integrates multi-objective reinforcement learning (MORL), model predictive control (MPC), adaptive particle swarm optimization (APSO), and Lyapunov stability theory into a unified adaptive system. The framework optimizes trajectory, offloading decisions, resource allocation, and CPU frequency control in highly dynamic IoT environments. Key strengths include:

- high task completion rate (94.5%),
- improved energy efficiency and load balancing,
- dynamic adaptability to bursty traffic,
- strong scalability across multiple UAVs.

Despite its strong performance, MOALF-UAV-MEC has limitations: high computational cost (due to MPC), complex implementation requirements (e.g., SDN support), and reliance on simulation rather than real-world experiments. These constraints suggest the need for more lightweight or practically deployable MEC optimization frameworks.

F. Task Completion Maximization in MEC Under Time and Energy Constraints

The work in highlights a critical oversight in existing MEC literature: while most studies minimize latency, energy, or cost, they rarely optimize the number of tasks actually completed within deadlines and energy limits. The authors propose a Mayfly Genetic Algorithm (MGA), combining genetic algorithms for global exploration with mayfly dynamics for fast convergence [2].

The algorithm jointly optimizes CPU frequency, offloading decisions, offloading ratios, and edge server resource distribution. Simulation results demonstrate that MGA:

- significantly increases task completion count,
- reduces energy consumption and cost,
- adapts effectively to user energy constraints,
- scales well to systems with up to 100 users.

Limitations include higher completion time in some scenarios, increased metaheuristic overhead for large populations, and simplified assumptions (e.g., one task per user). Still, the work contributes an important objective—maximizing completed tasks—that most MEC studies overlook.

G. DRL-Based UAV-MEC for High-Reliability Low-Latency IIoT

Another related work proposes a deep reinforcement learning scheme using Proximal Policy Optimization (PPO) [1] to achieve ultra-reliable low-latency communication (URLLC) in UAV-assisted IIoT networks . The UAV serves as a mobile redundant MEC node, improving reliability by compensating for degraded BS links.

The PPO agent controls UAV motion, device scheduling, and offloading decisions, receiving rewards for tasks meeting both reliability and latency constraints. Major advantages include:

- highest number of served devices among baselines,
- strong performance under 0.995–0.999 reliability targets,
- intelligent energy-aware trajectory adaptation,
- robustness to variations in device density.

However, the method is limited by discrete action space, sensitivity to hyperparameters, lack of multi-UAV evaluation, and the assumption that the UAV must pause movement to process tasks [1]. It remains simulation-based and does not explore continuous control trajectories.

H. Summary and Research Gap

In summary, prior literature presents strong foundations in UAV trajectory optimization, MEC offloading design, and machine-learning-based adaptation. However, existing works typically address only isolated aspects of the UAV-MEC problem, rely on restrictive assumptions, or incur high computational overhead. Even advanced frameworks like MOALF-UAV-MEC, MGA-based optimization, and PPO-driven IIoT MEC exhibit limitations in scalability, real-time feasibility, or generality.

III. System Model

M IoT devices are randomly located within a square plane of area A. The coordinates of device m are denoted by (x_m, y_m, z_m). Devices generate computation tasks according to independent Poisson processes with mean arrival rate λ_m.

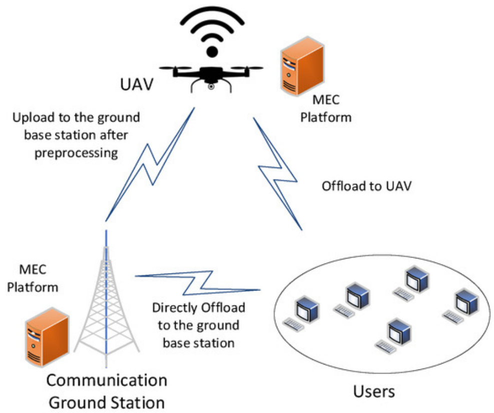

Fig. 1. Example of an ABS-aided MEC Wireless Network

Each task k is generated at time t_k^{gen}, has an input size D_k (bits), requires C_k CPU cycles to execute, and must be completed before its deadline t_k^{deadline}. The slack time of task k is defined as

$$t_k^{\text{slack}} := t_k^{\text{deadline}} - t_k^{\text{gen}}.$$

One ABS operates with a time-varying position $(x_n(t), y_n(t), z_n(t))$ and velocity $v(t)$. The ABS provides uplink connectivity and mobile edge computing services. To avoid confusion with the terrestrial base station, the ABS is referred to as the UAV hereafter. A far-away terrestrial base station (BS) provides centralized computing services.

The UAV uses Time Division Multiple Access (TDMA); therefore, at most one device transmits at any given time. For each scheduled transmission, the UAV allocates the full bandwidth B_{total} to the active device. No bandwidth splitting is performed.

For each task k generated by device m, one of the following three service options is selected:

(1) Local computation: The task is processed locally at the device. The required service time is

$$t_k^{\text{local}} := \frac{C_k}{f_m^{\text{IoT}}},$$

where f_m^{IoT} is the CPU frequency of device m.

(2) UAV offloading: The task is uploaded to the UAV and executed at the UAV MEC node. The total service time consists of uplink transmission and computation:

$$t_k^{\text{UAV}} := \frac{D_k}{R_m^{\text{UAV}}} + \frac{C_k}{f^{\text{UAV}}},$$

where R_m^{UAV} is the uplink rate to the UAV and f^{UAV} is the UAV CPU frequency. The downlink transmission time is assumed negligible due to the small output size.

(3) Terrestrial BS offloading: Similarly, offloading to the terrestrial BS incurs

$$t_k^{\text{BS}} := \frac{D_k}{R_m^{\text{BS}}} + \frac{C_k}{f^{\text{BS}}},$$

where R_m^{BS} and f^{BS} denote the uplink rate to the BS and its CPU frequency, respectively.

For both UAV and BS uplinks, the achievable transmission rate of device m is modeled as

$$R_m^{\text{UAV or BS}} := B_{\text{total}} \log_2\left(1 + \frac{P_m^{\text{r}}(d)}{\sigma^2}\right),$$

where $P_m^{\text{r}}(d)$ is the received signal power at distance d, and σ^2 is the noise power.

The received signal power is computed using the Okumura–Hata path loss model. All power quantities are expressed in dBm. The received power is given by

$$P_r(d) := P_T - L(d),$$

where the large-scale path loss is

$$\begin{aligned} L(d) &:= L_u(d) - 4.78[\log_{10} f]^2 + 18.33 \log_{10} f \\ &\quad - 40.94, \\ L_u(d) &:= 69.55 + 26.16 \log_{10} f - 13.82 \log_{10} z_n \\ &\quad - C_H(f, z_m) + (44.9 - 6.55 \log_{10} z_m) \log_{10} d, \\ C_H(f, z_m) &:= 0.8 + [1.1 \log_{10} f - 0.7] z_m - 1.56 \log_{10} f. \end{aligned}$$

The Okumura-Hata model is selected to capture distance, frequency, and antenna-height-dependent large-scale attenuation and is appropriate for open-area propagation.

Binary offloading variables are introduced:

$$\delta_k^{\text{local}},\ \delta_k^{\text{BS}},\ \delta_k^{\text{UAV}} \in \{0, 1\}, \qquad \delta_k^{\text{local}} + \delta_k^{\text{BS}} + \delta_k^{\text{UAV}} = 1.$$

The total serving time of task k is thus

$$t_k^{\text{serve}} = \delta_k^{\text{local}} t_k^{\text{local}} + \delta_k^{\text{BS}} t_k^{\text{BS}} + \delta_k^{\text{UAV}} t_k^{\text{UAV}}.$$

A task is considered successfully served if it meets its deadline constraint:

$$t_k^{\text{serve}} \le t_k^{\text{slack}}.$$

IV. Methodology

A. Problem Formulation

The objective of the proposed framework is to maximize the number of successfully completed computational tasks in a mobile edge computing network where IoT devices can offload their tasks to either a ground base station or an aerial UAV. The challenge is twofold: determining optimal offloading decisions for each task while simultaneously planning the UAV's trajectory to maximize network coverage and task completion rates.

The resulting problem becomes

$$\max_{\{\delta_k, x(t), v(t)\}} \quad \sum_k I_k^{\text{complete}} \tag{1}$$

$$\text{s.t.} \quad I_k^{\text{complete}} = \begin{cases} 1, & t_k^{\text{serve}} \le t_k^{\text{slack}} \\ 0, & \text{otherwise} \end{cases} \tag{C1}$$

$$\delta_k^{\text{local}} + \delta_k^{\text{BS}} + \delta_k^{\text{UAV}} = 1, \quad \forall k \tag{C2}$$

$$\sum_k a_k(t) \le 1, \quad \forall t \tag{C3}$$

$$x(t+1) = x(t) + v(t)\Delta t, \quad \forall t \tag{C4}$$

$$\|v(t)\| \le v_{\max}, \quad \forall t. \tag{C5}$$

$$P^{\text{success}}(\text{SNR}) \geq P^{min} \quad \text{(C6)}$$

(C1) defines the binary task completion indicator. A task is counted as successfully completed if and only if its total serving time does not exceed its available slack time.

(C2) enforces that each task is processed using exactly one service mode: local computation, UAV offloading, or terrestrial BS offloading.

(C3) represents the TDMA multiple-access restriction, which guarantees that at most one device can transmit to the UAV at any given time slot.

(C4) describes the motion of the UAV, where its position evolves according to its instantaneous velocity.

(C5) limits the UAV speed to a maximum allowable value.

(C6) states a channel is reliable to transmit over if the data can successfully traverse the link with probability 95%.

The objective represents the total number of tasks completed within their deadline; hence, late or partially served tasks do not contribute. This motivates scheduling and trajectory decisions that prioritize urgent tasks.

This optimization problem is mixed-integer due to the binary indicators σ_k, I_k^{complete}, and $a_k(t)$. It is also nonlinear because the achievable rate depends logarithmically on the received power, which in turn depends on the UAV-device geometry. The problem is dynamic because the UAV trajectory and scheduling decisions are time-indexed. Thus, the problem is NP-hard, which warrants an algorithmic approach to solving it.

The solution uses a two-layer optimization method that decomposes the complex problem into trajectory planning and task scheduling subproblems. First, The UAV's trajectory is optimized over the mission duration using two gradient-based strategies.

Online Optimization (Receding Horizon Control): At each decision point, the UAV looks ahead over a future time window (queued tasks.) The trajectory for this limited horizon is optimized using iterative gradient descent. The UAV only commits to the first step of the optimized trajectory. This process repeats at each time step, continuously adapting to new task arrivals.

Offline Optimization (Batch Planning): Before the mission, tasks are generated according to their expected arrival pattern. The entire trajectory spanning the mission duration is optimized jointly. Gradient descent iteratively refines UAV positions to maximize total task completions, and the complete trajectory is then executed as planned. This approach achieves better global optimization at the cost of adaptability the online method is capable of.

The solution also involves two control cases for comparison: a simple circular trajectory serves as a naive solution, and a "no UAV" scenario uses only the terrestrial base station to demonstrate the UAV's added value.

Second, a greedy scheduling algorithm determines the optimal processing location for each arriving task. It evaluates the three options of local processing, UAV offloading, and base station offloading. Among the feasible options meeting reliability and deadline constraints, the device selects the one minimizing total service time. If no option satisfies all constraints, the task is rejected as infeasible.

B. Simulation

The simulated network models a UAV-assisted MEC system with the following configuration:

- A baseline network of 5 IoT devices, one UAV, and one terrestrial base station. IoT devices are randomly distributed within an annular region of radius 50–350 m. The base station is located at $(700, 700)$ m, sufficiently far from the device cluster.
- Time is discretized into 70 steps over a total mission duration of 20 s, yielding $\Delta t \approx 0.286$ s.
- Tasks arrive according to a Poisson process with mean rate $\lambda = 0.5$ tasks/s per device.
- Uplink transmission uses the total bandwidth of 15 MHz due to TDMA.
- A transmission is considered feasible only if the channel success probability exceeds 95%.
- Computing resources are configured as follows: the UAV operates at a CPU frequency of 5 GHz, flies at a fixed altitude of 50 m, and is constrained to a maximum velocity of 20 m/s. The base station operates at 8 GHz with a height of 30 m. IoT devices operate at 1 GHz with an antenna height of 1.5 m.
- Task characteristics follow uniform distributions: data size is sampled from $[1.5, 7]$ Mb, computational density from $[500, 1500]$ cycles/bit, and task deadline from $[0.7, 2.0]$ s after generation.

The trajectory and offloading strategies described previously are implemented using the following settings:

- **No UAV:** Tasks are executed either locally or offloaded to the distant base station. This scenario serves as a baseline to quantify the benefit of UAV deployment.
- **Circular Baseline:** The UAV follows a fixed circular trajectory with radius 200 m centered at $(100, 100)$ m, without any trajectory optimization.
- **Online Optimization:** The UAV trajectory is optimized using Model Predictive Control (MPC) with a receding horizon of 15 time steps. The optimization is solved using the Adam optimizer in PyTorch with a learning rate of 2.0 and 20 iterations per time step.
- **Offline Optimization:** The UAV trajectory is optimized over the full mission horizon of $T = 70$ time steps using the Adam optimizer with a learning rate of 0.5. A total of 25 optimization iterations are performed.

V. Results

A. Trajectory Visualization and Qualitative Analysis

Before presenting the statistical performance results, the UAV trajectories produced by the different strategies are discussed. These visualizations provide qualitative insight into how trajectory design influences coverage and hence task completion rate.

Fig. 2 shows the fixed circular baseline, which does not adapt to device locations or task urgency. In contrast, the online MPC trajectory in Fig. 3 moves toward regions with higher instantaneous task demand, reflecting short-term urgency-aware control. The offline optimized trajectory in

Fig. 4 demonstrates globally coherent trajectory planning, with smoother transitions and reduced redundant motion. These visual differences provide an intuitive explanation for the performance gains observed in the subsequent quantitative sweeps.

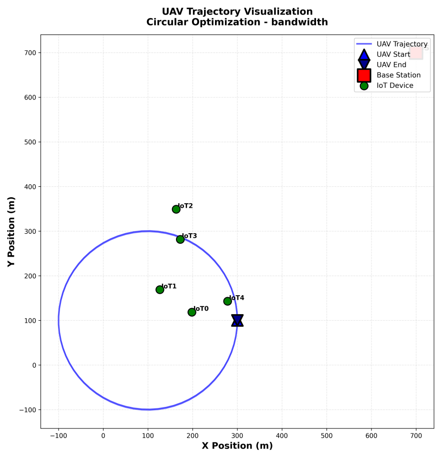

Fig. 2. UAV Trajectory Under the Circular Baseline Strategy

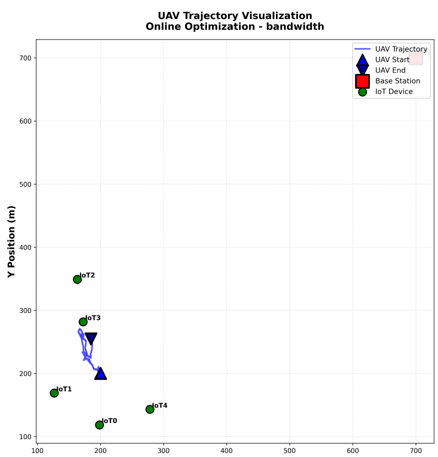

Fig. 3. UAV Trajectory Under the Online MPC-Based Optimization Strategy

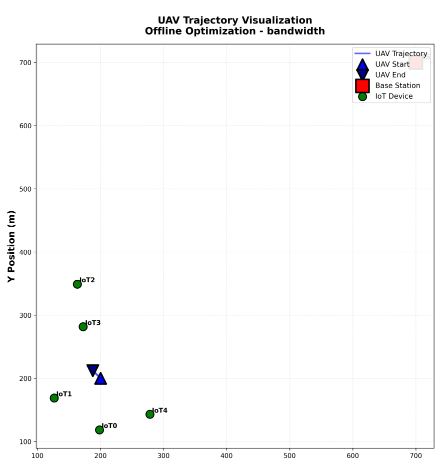

Fig. 4. UAV Trajectory Under the Offline Batch Optimization Strategy

To evaluate the network performance, Monte Carlo simulations are used to ensure statistical reliability. Each experimental configuration was repeated independently 5 times with different random seeds, resulting in a total of 260 simulation runs across all parameter sweeps and trajectory strategies. Three parameter sweeps were conducted to assess system sensitivity and scalability.

B. Task Data Size Sweep

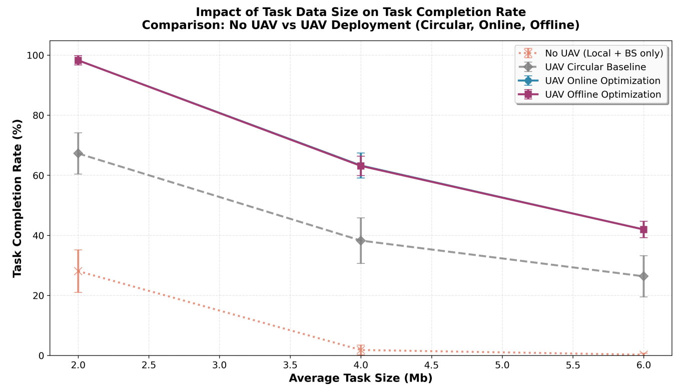

Fig. 5. The Variation of the Task Completion Rate Relative to Task Data Size

The average task size was varied over the ranges $(1, 3)$, $(2, 6)$, and $(3, 9)$ Mb (min, max). As shown in figure 5, both optimized trajectory strategies consistently outperform the circular and no-UAV baselines. As task sizes increase, communication latency becomes the dominant bottleneck, making trajectory-aware UAV placement more effective in reducing path loss and upload time. Conversely, for smaller tasks, the computational advantage of the terrestrial base station becomes more pronounced, narrowing the performance gap.

According to the benchmark suite output, the network benefits from a +33.9% improvement to the task completion rate by deploying the UAV. Compared to the circular baseline, the offline-optimized and online-optimized trajectories provide a +23.8% and +23.9% improvement to the task completion rate respectively.

C. Network Scalability Sweep

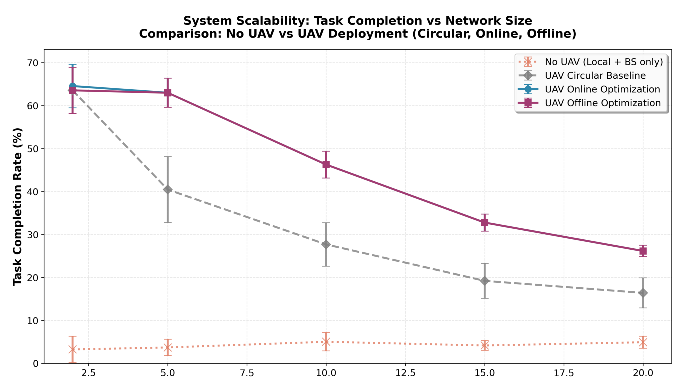

Fig. 6. The Variation of the Task Completion Rate Relative to the Number of Devices

The number of IoT devices was varied as $M \in \{2, 5, 10, 15, 20\}$. Overall performance degrades with increasing network load due to TDMA contention and limited UAV coverage, as the UAV cannot simultaneously serve many spatially distributed devices. Despite this degradation, both optimized trajectory methods consistently achieve the highest number of completed tasks across all network sizes, demonstrating improved scalability relative to the baseline strategies.

According to the benchmark suite output, the network benefits from a +29.3% improvement to the task completion rate by deploying the UAV. Compared to the circular baseline, the offline-optimized and online-optimized trajectories provide a +12.9% and +13.1% improvement to the task completion rate respectively.

D. Bandwidth Budget Sweep

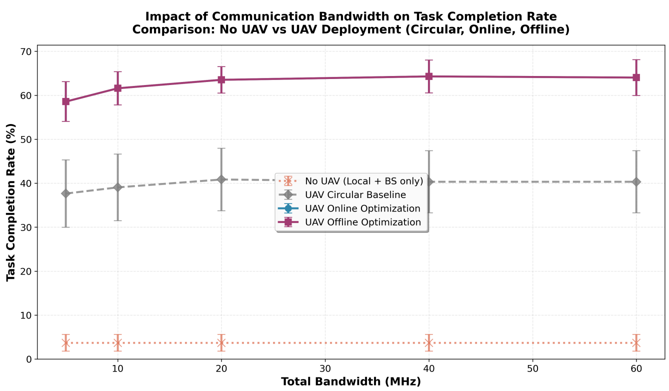

Fig. 7. The Variation of the Task Completion Rate Relative to the Bandwidth Budget

The total system bandwidth was varied as $B \in \{5, 10, 20, 40, 60\}$ MHz. Performance initially improves with increasing bandwidth but exhibits diminishing returns beyond a moderate threshold. This saturation behavior indicates that, under the considered settings, the network transitions from a communication-limited regime to a computation-limited regime, where additional bandwidth no longer yields significant gains in task completion.

According to the benchmark suite output, the network benefits from a +35.9% improvement to the task completion rate by deploying the UAV. Compared to the circular baseline, both the offline-optimized and online-optimized trajectories provide a +22.8% improvement to the task completion rate.

VI. Conclusion

This paper investigated trajectory optimization for UAV-assisted mobile edge computing networks with the objective of maximizing the number of completed computation tasks under deadline and compute resource constraints. A system model was developed that captures task generation, offloading decisions, wireless link dynamics, and UAV mobility. The resulting optimization problem was shown to be mixed-integer, nonlinear, and time-coupled, motivating the use of gradient-based trajectory optimization.

Two optimization strategies were evaluated: an online MPC-based approach and an offline batch optimization approach. Simulation results demonstrate that both optimized strategies perform similarly but significantly outperform fixed-trajectory and no-UAV baselines across all evaluated sweeps. Scaling the network further confirms that optimized trajectories maintain consistent performance advantages, and the bandwidth sweep reveals a transition from communication-limited to computation-limited operation, beyond which additional bandwidth offers diminishing returns.

These results confirm that trajectory-aware UAV deployment is a key enabler for reliable and low-latency MEC in dynamic IoT environments. Future work will extend this work to include the UAV's energy in the model, and simulate scenarios with multiple UAVs.

References

[1] M. A. Zahraa Sweidan, Sanaa Sharafeddine, "Rl-based mobile edge computing scheme for high reliability low latency services in uav-aided iiot networks," *Ad Hoc Networks*, vol. 166, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870524002579

[2] H. Yu, J. Liu, C. Deng, C. Chen, and K. Li, "Maximizing the number of completed tasks in mec considering time and energy constraints," *Soft Computing*, vol. 27, 2023.

[3] A. A. AL-Bakhrani, M. Li, M. S. Obaidat, and G. A. Amran, "Moalf-uav-mec: Adaptive multiobjective optimization for uav-assisted mobile edge computing in dynamic iot environments," *IEEE Internet of Things Journal*, vol. 12, 2025.