2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring) | 979-8-3503-1114-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/VTC2023-Spring57618.2023.10200917

MEC-assisted Low Latency Communication for Autonomous Flight Control of 5G-Connected UAV

Sourabh Solanki*, Asad Mahmood*, Vibhum Singh*, Sumit Gautam†,*, Jorge Querol*, and Symeon Chatzinotas*

*Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg
†Department of Electrical Engineering, Indian Institute of Technology Indore, India

Email:{sourabh.solanki, asad.mahmood, vibhum.singh, jorge.querol, symeon.chatzinotas}@uni.lu, sumit.gautam@iiti.ac.in

***Abstract*—Proliferating applications of unmanned aerial vehicles (UAVs) impose new service requirements, leading to several challenges. One of the crucial challenges in this vein is to facilitate the autonomous navigation of UAVs. Concretely, the UAV needs to individually process the visual data and subsequently plan its trajectories. Since the UAV has limited onboard storage constraints, its computational capabilities are often restricted and it may not be viable to process the data locally for trajectory planning. Alternatively, the UAV can send the visual inputs to the ground controller which, in turn, feeds back the command and control signals to the UAV for its safe navigation. However, this process may introduce some delays, which is not desirable for autonomous UAVs' safe and reliable navigation. Thus, it is essential to devise techniques and approaches that can potentially offer low-latency solutions for planning the UAV's flight. To this end, this paper analyzes a multi-access edge computing aided UAV and aims to minimize the latency of the task processing. More specifically, we propose an offloading strategy for a UAV by optimally designing the offloading parameter, local computational resources, and altitude of the UAV. The numerical and simulation results are presented to offer various design insights, and the benefits of the proposed strategy are also illustrated in contrast to the other baseline approaches.**

I. Introduction

The past few years have witnessed significant research attention for the development of unmanned aerial vehicles (UAVs) and their applications in academia and industries. In the envisioned 6G non-terrestrial networks (NTN), UAVs are going to play a prominent role [1] owing to the multiple new applications such as aerial relays, flying base stations (BSs), pollution monitoring, and internet-of-things (IoT). The potential of UAV applications is also evident by the ongoing standardization efforts from 3GPP, IEEE, and ITU [2]. Traditional applications of UAVs have a limited range of communication due to the adoption of short-range network access technologies such as WiFi. To enhance this range, cellular-connected UAVs are being extensively researched where a UAV is considered to be connected to a radio access network (RAN) such as LTE or 5G. For example, 3GPP in its Release 15 studied LTE-supported UAVs, while Releases 17 and 18 discuss 5G enhancements for UAVs [2]. To further advance UAV applications, safe and reliable UAV navigation beyond visual line-of-sight (LoS) is critical. For beyond visual LoS (BVLoS) navigation, it requires the UAV to autonomously govern its trajectory, which in turn requires the constant visual inputs of the flying environment. In particular, with the help of an onboard camera, image/video data can be exploited to identify any obstacles. Moreover, UAV can receive real-time inputs from ground control about the new no-fly zones based on which UAV is required to update the trajectory by implementing dynamic geofencing algorithms. However, all these tasks require extensive processing and storage capabilities at UAV which eventually increase the power consumption and hence reduce the flight time. To address this issue, such complex task processing can be offloaded to multi-access edge computing (MEC) servers which subsequently send the processed data back to UAV [3].

MEC, on the other hand, is an emerging technology that can bring cloud facilities near the edge of a network that can provide efficient and rapid data processing capabilities with energy efficiency [4]. As such, the application of MEC is particularly useful for delay-sensitive services, including UAV flight control. In fact, one of the crucial challenges for a UAV includes its safe and reliable operation in the highly regulated air-space. Therefore, autonomous flight control of UAV becomes a time-sensitive mechanism to ensure compliance with the strict guidelines of the aviation authorities. To this end, the MEC support can enable the low latency solution for UAV's autonomous flight control.

Various research works have focused on leveraging MEC technique in the context of UAV networks [5]–[11]. In [5], the authors introduced a cloudlet mounted on a UAV for MEC operations where mobile users with limited processing abilities on the ground utilize the computational resources of the UAV for computational offloading. For such a design, the joint optimization of bit allocation and path planning of UAV is tackled based on successive convex approximation methods. The authors in [6] analyzed a UAV-MEC system and studied joint resource and workflow scheduling for IoT networks. It has been considered that a UAV first powers the IoT devices and then those devices send the sensed data to a UAV for computation and processing. Elie *et al.* in [7] investigated UAV-aided computation offloading for IoT while considering the stringent requirements of latency and reliability to optimize UAV placement, offloading decisions, and radio and computational resources. A joint resource allocation and trajectory optimization problem has been examined in [8] where multiple UAVs are exploited as a parallel computational server for offloading the data from IoT users. Further, work in [9] optimized the latency for a UAV-enabled MEC system where UAVs are considered to cache, process, and deliver virtual reality content. The authors in [10] considered a learning-based framework for resource allocation in UAV-MEC for industrial IoT systems. In [11], joint resource and trajectory optimization is studied to address security concerns in UAV-MEC systems.

Common to all the above works is that they considered edge computation at UAVs. However, for some applications, it may not be viable for UAVs to be equipped with computational resources-rich hardware owing to limited onboard space, power consumption, and flight time. To address this issue, some work considered the offloading of UAV data to a terrestrial MEC server [12]–[14]. For example, in [12], the authors considered an MEC offloading scenario, especially focusing on the security issue of the physical layer. Ye *et al.* in [13] studied an ad

979-8-3503-1114-3/23/$31.00 ©2023 IEEE

hoc network scenario where a UAV first collects the data from the ground users and then transmits it to the MEC server for offloading. The authors in [14] attempted to maximize the energy efficiency of the UAV, which is offloading task to the terrestrial MEC unit, by optimizing the UAV's mobility and computation dynamics. Nevertheless, all these works considered LoS dominant large-scale fading for their analyses while ignoring the small-scale fading which may arise in dense urban scenarios due to multipath from high-rise buildings and other flying vehicles [15].

Motivated by the aforementioned discussion and different from the existing works, in this paper, we investigate a new application of MEC-assisted UAV for autonomous flight control. More specifically, it is considered that a UAV is equipped with cameras to acquire visuals from its surroundings. By processing the visual data, the UAV can identify the potential obstacles and accordingly its trajectory can be planned. However, due to limited onboard processing capability, the UAV relies on an MEC server at the edge to process visual input. Based on the received processed data, UAV can autonomously update its trajectory by constantly monitoring the surrounding environment. Nevertheless, for real-time trajectory planning and UAV's safe navigation, processing, offloading, and computation delay have to be minimized. To this end, this work analyzes a MEC-aided UAV and aims to minimize the latency for task processing. Importantly, we propose an offloading strategy for a UAV by optimally designing the offloading parameter, local computational resources, and altitude of UAV. The proposed strategy is also compared with the other baseline approaches such as local computation, edge computation, and binary offloading through the numerical and simulation results.

II. System Description

We consider a 5G-connected UAV system for autonomous flight control. A representative illustration of the application is shown in Fig. 1. Herein, a UAV is connected to a ground control station using a gNB and is responsible for defining the initial trajectory or waypoints. Ground control is necessary for UAV applications for safety and emergency situations. It is assumed that the UAV's flight mission includes fixed starting and endpoints. UAV needs to autonomously figure out the appropriate trajectory for navigating through these points. For this, the UAV is equipped with a camera/sensors to obtain visual inputs from the surrounding environment. Visual data are then processed locally at the UAV and at the edge to obtain the new trajectory. To enable such an application for 5G-connected UAV, we address the problem of latency minimization for MEC offloading and computation design. Since we aim to optimally allocate the local computational resources at UAV while also designing the offloading policy, we explicitly focus on the uplink transmission between UAV and MEC server. UAV is also capable of some computational processing as it can be equipped with a lightweight processor [14]. In contrast, MEC server has abundant computational resources and transmission capabilities; therefore, the time consumed for edge processing and downlink uploading is assumed to be negligible compared to the local computation time [16]. This is also based on the reasoning that the processed data size (new waypoints for the trajectory) can be significantly smaller than the input task (visual data) and thus, transmission time from a sophisticated MEC server would take negligible time. Moreover, as widely adopted in the literature [12], [13], MEC and gNB are assumed to be co-located. Further, it is considered that the data is processed by parallel computing, i.e., local and edge computations happen simultaneously. For

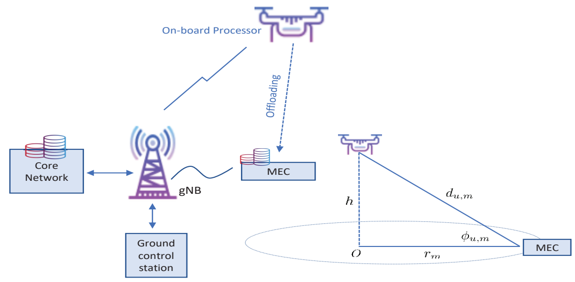


Fig. 1: System model.

uplink offloading, the UAV is considered to be equipped with a single antenna due to onboard space limitations. Furthermore, after acquiring sufficient visual inputs, UAV can hover at a fixed location and is able to adjust its altitude for data offloading to obtain new trajectories for minimizing the latency cost towards computation and processing.

A. Channel Model

To characterize the air-to-ground (A2G) propagation, we adopt a model that accounts for both large-scale and small-scale fading. In particular, for the UAV, the large-scale fading model depends on the altitude, distance, and elevation angle, which are dominant factors for A2G propagation. As such, the elevation angle $\phi_{u,m}$ between the UAV and the MEC server can be expressed as $\phi_{u,m} = \arctan\left(\frac{h}{r_m}\right)$, with h and r_m being, respectively, the UAV's adjustable altitude and the horizontal distance between the UAV's projection on the ground and MEC, as shown in Fig. 1. Further, based on the elevation angle, the LoS probability between the UAV and the MEC is given by [17]

$$\mathbb{P}(\phi_{u,m}) = \frac{1}{1 + \mathcal{C}\exp\left(-\mathcal{B}(\phi_{u,m} - \mathcal{C})\right)}, \tag{1}$$

where $\mathcal{B}$ and $\mathcal{C}$ are environment-dependent constants. Following the above formulations, the path-loss exponent can be given by [18], [19] $\alpha(\phi_{u,m}) = \mathbb{P}(\phi_{u,m})e_u + f_u$, where e_u and f_u are constants that depend on the propagation environment i.e., urban, suburban, dense urban, etc. The channel coefficient $g_{u,m}$ between the UAV and the MEC is assumed to follow Nakagami-*m* distribution to characterize the small-scale fading. Accordingly, the channel gain follows the Gamma distribution with probability density function (PDF) as $f_{|g_{u,m}|^2}(x) = \left(\frac{m_{u,m}}{\Omega_{u,m}}\right)^{m_{u,m}} \frac{x^{m_{u,m}-1}}{\Gamma(m_{u,m})} \exp\left(-m_{u,m}x\right)$, where $m_{u,m}$ is a fading severity parameter and $\Omega_{u,m}$ is the average fading power [20]. Nakagami-*m* distribution is a generalized model and can represent various fading scenarios. For instance, it can also encapsulate the Rician fading by setting its parameter as $m_{u,m} = \left(1 - \left(\frac{K}{k+1}\right)^2\right)^{-1}$ [21], where K is a Rician factor. Note that such a channel model incorporating large- and small-scale fading is commonly adopted in the literature [9].

B. Computation Model

We consider a computation model based on the partial offloading strategy. In such an approach, the UAV can exploit its computational resources to compute the data locally while also leveraging the resource-rich terrestrial MEC. More specifically, it is assumed that the UAV offloads the β portion of the data to the MEC server while the remaining $(1-\beta)$ is computed locally. We consider the full granularity in data partitioning such that the data can be divided into any subset size [22]. Following the dynamic voltage scaling (DVS) model [22], the execution time for the local computation T_{UL} at the UAV and the offloading time T_{UO} can be, respectively, given by [13], [16]

$$T_{\text{UL}} = \frac{(1-\beta)c}{f_{\text{LC}}} \text{ and } \quad T_{\text{UO}} = \frac{\beta X_b}{B\log_2\left(1+\Lambda_{u,m}\right)}, \quad (2)$$

where c represents the required central processing unit (CPU) cycles to process the X_b bits of task, f_{LC} denotes the UAV's CPU frequency in cycles per second. Let $\Lambda_{u,m} = P_o d_{u,m}^{-\alpha(\phi_{u,m})} \frac{|g_{u,m}|^2}{\sigma_o^2}$, with $\sigma_o^2 = BN_o$ being the variance of additive white Gaussian noise (AWGN), N_o is noise power spectral density and P_o denotes the offloading power.

C. Energy Consumption

The total energy consumption of UAV can be written as [23]

$$E_{u,p} = \int_t^{t+T_p} P_{u,p}\left(v(t)\right)dt + \frac{1}{2}m_u\left(v^2(t+T_p) - v^2(t)\right) + m_u g\left(h(t+T_p) - h(t)\right) + P_c T_p. \quad (3)$$

In (3), the first term corresponds to propulsion-related consumption while the second and third terms indicate kinetic and potential energy, respectively. The last term accounts for communication-related consumption. The propulsion power $P_{u,p}$ in 3 can be modeled by [24]

$$P_{u,p}\left(v(t)\right) = P_0\left(1 + \frac{3v^2(t)}{u_{\text{tip}}^2}\right) + \frac{1}{2}d_0\rho s A v^3(t) + P_i\left(\sqrt{1+\frac{v^4(t)}{4v_0^4}} - \frac{v^2(t)}{2v_0^2}\right)^{\frac{1}{2}}, \quad (4)$$

where $v(t)$ is the velocity of the UAV, while the other modeling parameters in (4) are dependent on the weight of the UAV, air density, and rotor disc area, as defined in [24]. Moreover, the power consumption for the local computation at UAV can be given by [22] $P_{\text{LC}} = k f_{\text{LC}}^3$, where k is a coefficient whose value depends on the chip architecture. Hence, the energy consumption to compute the $(1-\beta)X_b$ bits locally can be calculated, using the execution time T_{UL} in (2), as

$$E_{\text{LC}} = c(1-\beta)k f_{\text{LC}}^2. \quad (5)$$

Remark: From T_{UL} in (2), it can be witnessed that increase in local computation frequency f_{LC} can reduce the local computation time. In contrast, the rise in f_{LC} leads to increased energy consumption, as deduced from (5). Therefore, there exists an underlying trade-off between the computation time and energy consumption; thus, it is crucial to optimize the f_{LC} under the limited energy budget of UAV.

III. Problem Formulation

Since parallel computing has been considered for task processing, based on the local computation time T_{UL} and the offloading time T_{UO} in (2), the total latency cost towards the computation can be given by

$$T_{\text{L}} = \max\left(T_{\text{UL}}, T_{\text{UO}}\right). \quad (6)$$

For the autonomous control of UAV, the delay has to be minimized while optimally designing the offloading, computational resources, and UAV placement. More specifically, we minimize the total latency T_{L} by jointly optimizing the offloading parameter β, local computation frequency f_{LC}, offloading power P_o, and UAV's altitude h. To this end, the resulting problem can be formulated as

$$(\mathbf{P}): \min_{\beta, f_{\text{LC}}, P_o, h} \max\left(\frac{c(1-\beta)}{f_{\text{LC}}}, \frac{\beta X_b}{B\log_2\left(1+\Lambda_{u,m}\right)}\right)$$

$$\text{s.t. } \frac{\beta P_o X_b}{B\log_2\left(1+\Lambda_{u,m}\right)} + c(1-\beta)k f_{\text{LC}}^2 \leq E_u - E_{u,p}, \quad (7a)$$

$$0 \leq \beta \leq 1, \quad (7b)$$

$$h_{\min} \leq h \leq h_{\max}, \quad (7c)$$

$$0 \leq f_{\text{LC}} \leq f_{\text{LC}}^{\max}, \quad (7d)$$

$$0 \leq P_o \leq P_o^{\max}. \quad (7e)$$

In the above formulation, the constraint (7a) represents the energy causality constraint which ensures that the energy consumption for the local computation, offloading, and propulsion is less than the total energy E_u at the UAV. The constraint (7b) determines the portion of data to be offloaded, and it takes values between 0 and 1. It is worth noting that the formulated problem also encapsulates the local computation ($\beta = 0$) and the full offloading ($\beta = 1$) as the special cases. The bound in (7c), where $h_{\min}$ and $h_{\max}$ represent the minimum and maximum altitudes of the UAV, restricts the altitude of the UAV to avoid conflict with terrestrial or other air-borne vehicles. Furthermore, $f_{\text{LC}}^{\max}$ and $P_o^{\max}$ in (7d) and (7e), respectively, denote the maximum computational frequency at UAV and peak offloading power. Note that all the parameters are conveyed to a central controller or cloud manager at gNB [12] which executes the optimization process and sends the optimal parameters to UAV and GCS via feedback or control links through adequate signalling.

IV. Proposed Solution

The problem ($\mathbf{P}$) appears to be non-convex with respect to the optimization variables, and thus its solution is not straightforward. To overcome this, we decouple the original optimization problem and iteratively solve it as follows.

A. Optimal Altitude of UAV

Recalling that after receiving the visual inputs, the UAV offloads/computes the data for processing to receive new trajectories. For this, the UAV at any given time instant hovers at a fixed location and is able to adjust its altitude within the permissible altitude range. Also, note that UAV can not manoeuvre horizontally to avoid any no-fly zones or obstacles before it is updated with the new waypoints. Therefore, we explicitly consider the vertical placement of UAV to minimize the latency. In a UAV communication network, the LoS probability is an important parameter that highly impacts the system's performance. As perceived from (1), the LoS probability is mainly determined by the altitude of the UAV, for a fixed horizontal distance. Apparently, the rise in altitude increases the angle of elevation, resulting in an increased LoS probability and hence reduced path loss. In other words, the optimal altitude of UAV can be obtained by minimizing the path-loss. Mathematically, this can be expressed as follows:

$$\min_h \quad d_{u,m}^{-(\mathbb{P}(\phi_{u,m})e_u + f_u)}, \quad (8)$$

$$\text{s.t.} \quad (7\text{c}) \quad (9)$$

The primary aim of (8) is to optimally place the UAV in such a way that the angle of elevation ensures the highest LoS probability leading to minimal path loss.

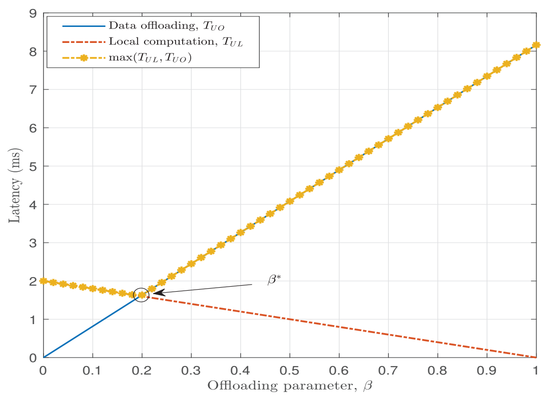


Fig. 2: Latency versus offloading parameter.

B. Resource Allocation

Following that, to deal with the non-convexity of the problem (**P**), we first apply the change of variable as $R_{u,m} = \log_2(1 + \Lambda_{u,m})$. For further simplification, we define another metric as: $\kappa(\beta, f_{\text{LC}}, R_{u,m}) = \max\left(\frac{c(1-\beta)X_b}{f_{\text{LC}}}, \frac{\beta X_b}{BR_{u,m}}\right)$. Subsequently, (**P**) can be reduced as

$$(\mathbf{P1}): \min_{\beta, f_{\text{LC}}, R_{u,m}} \kappa(\beta, f_{\text{LC}}, R_{u,m})$$

$$\text{s.t. } \frac{\beta X_b (2^{R_{u,m}} - 1) d_{u,m}^{\alpha(\phi_{u,m})} \sigma_o^2}{BR_{u,m}|g_{u,m}|^2} + c(1-\beta)k f_{\text{LC}}^2 X_b \le \tilde{E}_u, \tag{10a}$$

$$0 \le R_{u,m} \le R_{u,m}^{\max}, \tag{10b}$$

$$(7\text{b}), (7\text{d}), \tag{10c}$$

where $R_{u,m}^{\max} = \log_2\left(1 + P_o^{\max} d_{u,m}^{-\alpha(\phi_{u,m})} \frac{|g_{u,m}|^2}{\sigma_o^2}\right)$ and $\tilde{E}_u = E_u - E_{u,p}$. On carefully observing the objective function (**P1**), the latency can be minimized while keeping other variables fixed, by judiciously designing β such that $T_{\text{UL}} = T_{\text{UO}}$, i.e.,

$$\frac{c(1-\beta)}{f_{\text{LC}}} = \frac{\beta X_b}{BR_{u,m}}. \tag{11}$$

This observation can also be validated using Fig. 2. The parameters to plot this figure are given in Section V. As a consequence, optimal offloading parameter can be obtained as

$$\beta^* = \frac{cBR_{u,m}}{f_{\text{LC}} + cBR_{u,m}}. \tag{12}$$

On invoking β^* in (**P1**), it can be reformulated as

$$(\mathbf{P2}): \min_{f_{\text{LC}}, R_{u,m}} \frac{cX_b}{f_{\text{LC}} + cBR_{u,m}}$$

$$\text{s.t. } \frac{cX_b\left(2^{R_{u,m}} - 1\right) d_{u,m}^{\alpha(\phi_{u,m})} \frac{\sigma_o^2}{|g_{u,m}|^2}}{f_{\text{LC}} + cBR_{u,m}} + \frac{ck f_{\text{LC}}^3}{f_{\text{LC}} + cBR_{u,m}} \le \tilde{E}_u, \tag{13a}$$

$$0 \le R_{u,m} \le R_{u,m}^{\max} \tag{13b}$$

$$(7\text{d}). \tag{13c}$$

Further, minimization problem (**P2**) can be simply reduced to

$$(\mathbf{P3}): \max_{f_{\text{LC}}, R_{u,m}} f_{\text{LC}} + cBR_{u,m}$$

$$\text{s.t. } (7\text{d}), (13\text{a}), (13\text{b}). \tag{14}$$

Algorithm 1 Proposed Optimization Framework

Initialization: Define all parameters of the system and Ω_o, **error**.
while ***error*** $\le \epsilon$ **do**
  $h^* \leftarrow$ solve (8)
  // Given the value of all the height variables, next step is to find value of computation and communication resources
  $\Omega^* \leftarrow$solve **(P3)** to find $f_{\text{LC}}, R_{u,m}, h$
  // Calculate the error, **error** = $\Omega^* - \Omega_o$
Return $h^*, f_{\text{LC}}^*, R_{u,m}^*$

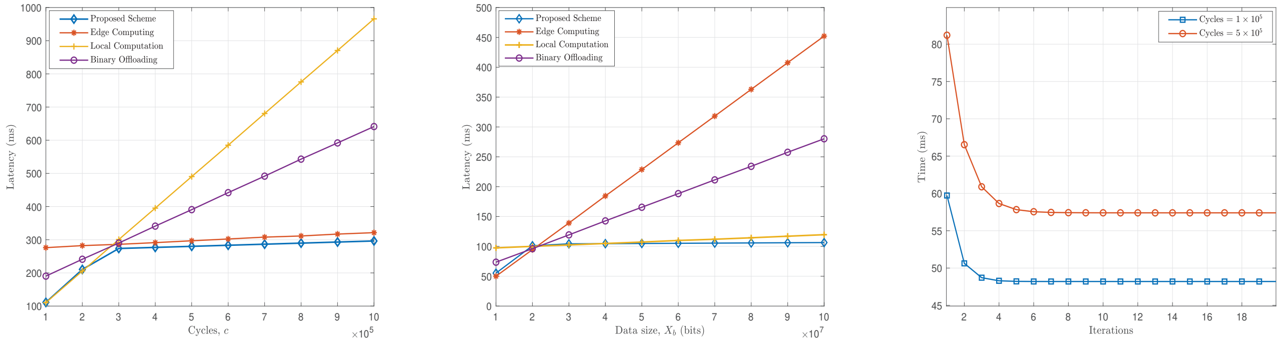
It is clear that (**P3**) is still non-convex while considering the joint optimization of f_{LC} and $R_{u,m}$. However, it is noteworthy that the objective (and the corresponding constraints) become convex while solely optimizing one variable and keeping the other fixed, and vice-versa. In this context, we propose an alternating iterative algorithm to solve (**P3**). The sub-problems, i.e., for the case where one parameter is optimized while keeping the other fixed, and vice-versa; can be solved iteratively with the help of the standard convex optimization solvers, such as CVX [25]. The overall solution is summarized as in Algorithm 1.

V. Numerical Results

In this section, we evaluate the performance of the considered system while employing the proposed algorithm using the numerical and simulation results. For this, various system parameters are set [5], [22] as $k = 10^{-28}$, $c = [1 \times 10^4, 10 \times 10^4]$, $B = 10$ MHz, $f_{\text{LC}}^{\max} = 500$ MHz, $\mathcal{C} = 0.5, \mathcal{B} = 20, X_b = [10, 100]$ Mb, $e_u = -1.5, f_u = 3.5, m_{u,m} = 3, \Omega_{u,m} = 1, r_m = 180$ m, $h_{\min} = 50$ m, $h_{\max} = 200$ m, $P_o^{\max} = -20$ dB, $N_o = -174$ dBm/Hz, $\tilde{E}_u = 1 \times 10^4$ Joules.

In Fig. 3(a), we demonstrate the latency performance against the number of cycles required at UAV for computation of X_b bits of data, i.e., c. In particular, we also compare the performance of our proposed strategy with the other baseline approaches such as 1) Edge computing; 2) Local computation; 3) Binary offloading. The case of edge computing accounts for the full offloading ($\beta = 1$) where all the data is computed at the edge. In contrast, the second case of local computation corresponds to $\beta = 0$ when the data is computed locally. In comparison, binary offloading considers the computation either locally or at the edge. As evident from the pertinent curves, the proposed scheme outperforms the other baseline approaches in terms of latency. Also, the gap between the proposed and any other schemes grows significantly as the number of cycle requirements increases. Essentially, for the lower c, the performance is closer to the local computation, while for the higher c, it is closer to the edge computation. This can be associated with the fact that higher cycle requirements would increase the computational latency of UAV, and thus, in that case, a large portion of data would be offloaded. Local computation is more favorable for the lower c as it consumes less time.

Fig. 3(b) examines the variations in latency against the computational data size while keeping the number of computation cycles c as fixed. Since the offloading time would be less for the lower amount of data, the performance of the proposed approach is closer to the edge computation. This is rational and intuitive since offloading to the edge in such a case can save computation time as the number of computation cycles required is kept fixed. Further, for the large X_b, the proposed scheme's performance is aligned with the local computation, as offloading a large amount of data is more time-consuming. Clearly, the proposed scheme outperforms the other approaches.



(a) (b) (c)

Fig. 3: (a) Latency versus number of required cycles; (b) Latency versus data size; (c) Convergence of algorithm.

Finally, we demonstrate the convergence of the proposed algorithm in Fig. 3(c). It can be seen that the algorithm's convergence time varies depending on the simulation parameters. The results show that the algorithm converges faster when the number of cycle requirements is low. On the other hand, more iterations are required to converge to the best solution for a large number of cycle requirements because of its extensive computation requirements. For instance, a five times increase in the number of cycles requirements leads to around a 20% increase in convergence time.

VI. Conclusion

We investigated a latency minimization problem for a MEC-assisted UAV system. Under the developed approach, the joint optimization of power, placement, and computational resources leads to efficient flight management of the UAV. In comparison with the other baseline approaches, it is shown that the proposed scheme can potentially minimize the latency while judiciously allocating the task to the edge or the local processor under the limited energy resources. In essence, the proposed framework can be potentially useful for enabling the autonomous-flight control of a 5G-connected UAV. For future work, the proposed approach can be extended to a more generalized scenario to study the impact of interference from other terrestrial and flying nodes.

Acknowledgment

This work was supported by the Luxembourg National Research Fund (FNR)-5G-Sky Project, ref. 13713801, and the SMC-funded Micro5G project.

References

[1] M. M. Azari *et al.*, "Evolution of Non-Terrestrial Networks from 5G to 6G: A Survey," *IEEE Communications Surveys Tutorials*, vol. 24, no. 4, pp. 2633–2672, 2022.

[2] M. M. Azari, S. Solanki, S. Chatzinotas, and M. Bennis, "THz-Empowered UAVs in 6G: Opportunities, Challenges, and Trade-offs," *IEEE Communications Magazine*, vol. 60, no. 5, pp. 24–30, 2022.

[3] Q. Chen *et al.*, "Edge Computing Assisted Autonomous Flight for UAV: Synergies between Vision and Communications," *IEEE Communications Magazine*, vol. 59, no. 1, pp. 28–33, 2021.

[4] T. Taleb *et al.*, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[5] S. Jeong, O. Simeone, and J. Kang, "Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2018.

[6] Y. Du *et al.*, "Joint Resources and Workflow Scheduling in UAV-Enabled Wirelessly-Powered MEC for IoT Systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10 187–10 200, 2019.

[7] E. E. Haber, H. A. Alameddine, C. Assi, and S. Sharafeddine, "UAV-Aided Ultra-Reliable Low-Latency Computation Offloading in Future IoT Networks," *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6838–6851, 2021.

[8] X. Qin, Z. Song, Y. Hao, and X. Sun, "Joint Resource Allocation and Trajectory Optimization for Multi-UAV-Assisted Multi-Access Mobile Edge Computing," *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1400–1404, 2021.

[9] A. A. Nasir, "Latency Optimization of UAV-Enabled MEC System for Virtual Reality Applications Under Rician Fading Channels," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1633–1637, 2021.

[10] L. Sun, L. Wan, and X. Wang, "Learning-Based Resource Allocation Strategy for Industrial IoT in UAV-Enabled MEC Systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5031–5040, 2021.

[11] Y. Xu, T. Zhang, D. Yang, Y. Liu, and M. Tao, "Joint Resource and Trajectory Optimization for Security in UAV-Assisted MEC Systems," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 573–588, 2021.

[12] T. Bai, J. Wang, Y. Ren, and L. Hanzo, "Energy-Efficient Computation Offloading for Secure UAV-Edge-Computing Systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6074–6087, 2019.

[13] Y. Yu *et al.*, "UAV-Aided Low Latency Multi-Access Edge Computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4955–4967, 2021.

[14] J. Zhou, D. Tian, Z. Sheng, X. Duan, and X. Shen, "Joint Mobility, Communication and Computation Optimization for UAVs in Air-Ground Cooperative Networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2493–2507, 2021.

[15] Y. Liu *et al.*, "UAV Communications Based on Non-Orthogonal Multiple Access," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 52–57, 2019.

[16] J. Park, S. Solanki, S. Baek, and I. Lee, "Latency Minimization for Wireless Powered Mobile Edge Computing Networks With Nonlinear Rectifiers," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8320–8324, 2021.

[17] S. Solanki, J. Park, and I. Lee, "On the Performance of IRS-Aided UAV Networks with NOMA," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 9038–9043, 2022.

[18] M. M. Azari, F. Rosas, K.-C. Chen, and S. Pollin, "Ultra Reliable UAV Communication Using Altitude and Cooperation Diversity," *IEEE Transactions on Communications*, vol. 66, no. 1, pp. 330–344, 2018.

[19] Y. Liu *et al.*, "UAV-Aided Wireless Power Transfer and Data Collection in Rician Fading," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3097–3113, 2021.

[20] U. Singh, S. Solanki, D. S. Gurjar, P. K. Upadhyay, and D. B. d. Costa, "Wireless power transfer in two-way af relaying with maximal-ratio combining under Nakagami-m fading," in *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, 2018, pp. 169–173.

[21] S. Solanki, S. Gautam, S. K. Sharma, and S. Chatzinotas, "Ambient Backscatter Assisted Co-Existence in Aerial-IRS Wireless Networks," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 608–621, 2022.

[22] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268–4282, 2016.

[23] Y. Zeng, Q. Wu, and R. Zhang, "Accessing From the Sky: A Tutorial on UAV Communications for 5G and Beyond," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2327–2375, 2019.

[24] Y. Zeng, J. Xu, and R. Zhang, "Energy Minimization for Wireless Communication With Rotary-Wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[25] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.