Edge Cloud Resource-aware Flight Planning for Unmanned Aerial Vehicles

Oussama Bekkouche[1], Tarik Taleb[1,2], Miloud Bagaa[1], and Konstantinos Samdanis[3]

[1]Communications and Networking Department, Aalto University, Finland. Email: firstname.lastname@aalto.fi
[2]Centre for Wireless Communications, University of Oulu, Finland. Email: tarik.taleb@oulu.fi
[3]Nokia Bell Labs, Munich, Germany. Email: konstantinos.samdanis@nokia.com

***Abstract*—Unmanned Aerial Vehicles (UAVs) can offer a plethora of applications, provided that the appropriate ground control and complementary computing and storage services are available in close proximity. To accomplish this, edge cloud platforms, deployed at or close to the base stations, are essential. However, current UAV travel planning does not take into account the resource constraints of such edge cloud platforms. This paper introduces an aligned process for UAV flight planning and networking resource allocation, minimizing the total traveled distance. It proposes two solutions, namely (i) a Multi-access Edge Computing (MEC)-Aware UAVs' Path planning (MAUP) based on integer linear programming and (ii) an Accelerated MAUP (AMAUP), i.e., a heuristic and scalable approach that adopts the shortest weighted path algorithm considering directed graphs. The performance of the two solutions are evaluated using computer-based simulations and the obtained results demonstrate the effectiveness of the two solutions in achieving their design goals.**

I. Introduction

Unmanned Aerial Vehicles (UAVs), also known as drones, are low-altitude flying aircrafts without a human pilot on board. UAVs rely on autonomous navigation combined with ground-based control [1]. Owing to their agility, low-cost, convenience in reaching certain remote areas and rapid delivery, UAVs are gaining momentum, increasing their market footprint beyond the initial military applications of surveillance and reconnaissance. UAVs can be harnessed in a wide range of public and civil applications, such as health care, agriculture, safety, transportation management and cargo delivery. Such applications require a Beyond Visual Line-of-sight (BVLOS) control and monitoring of UAVs. In this context, the 5^{th} Generation of mobile communications (5G) that introduces service-aware cloud orchestration is well placed to support ground control communications for UAVs during their missions.

Besides conventional services provided by mobile operators to User Equipment (UE) such as localization, authentication, and ready-for-use service connectivity using a licensed spectrum, UAVs can benefit from the latest advances in edge cloud technologies, such as Multi-Access Edge Computing (MEC) [2], [3]. By leveraging the benefits of MEC at the network edge (e.g., base stations, i.e. eNodeB or gNB), a dramatic decrease in the communication latency can be induced between the edge-hosted services and the connected devices (e.g., UAVs), enabling new service capabilities and applications. From the UAVs perspective, MEC can be used to host UAVs control services as discussed in [4] or to offload intensive computation (e.g., image processing) [5]. Indeed, UAVs are expected to feed complex Artificial Intelligence (AI) tasks, e.g. situation awareness for emergency services, with critical time of response. Meanwhile their on-board computation resources are very limited due to their size, weight, and power (SWAP) limitations [6]. To overcome these limitations, computation and storage services are offloaded from UAVs to the nearest available MEC platform.

However, unlike centralized cloud data-centers, resources at MEC nodes are limited and hence smart exploitation is encouraged. This is mainly achieved through optimal resource provisioning and service placement [7]. Whilst resource provisioning aims at finding the optimal amount of computation and storage resources to allocate at the edge nodes, service placement aims at finding the optimal distribution of a given service over a set of edge nodes in such a way that this service can be available for the maximum number of users. Consequently, UAVs cannot be served from a random MEC node within their coverage vicinity since resources at that node may be insufficient and/or the desired service may not be available.

In order to ensure a safe operation of UAVs in the airspace and ground, each UAV should use a regulated geo-fence, following a predefined flight path issued by the UAVs Traffic Management (UTM) system [8]. The flight planning process takes into consideration the state of the air traffic and the airspace restrictions, but cannot ensure the availability of edge resources and/or services along the flight trajectory. Hence, it is impossible to manage the MEC potential resource and service allocation needed in UAVs' applications, unless the UTM process is aligned with networking operations.

To address such an issue, this paper proposes two solutions to guarantee that a flying UAV will connect to base stations that allow to use MEC nodes with sufficient resources, while minimizing the total traveled distance. The first solution named MEC-Aware UAVs' Path planning (MAUP) uses integer linear programming to find the optimal flight path that satisfies the UAVs' connectivity and computation resource requirements. However, given its complexity, MAUP may take long to find an optimal flight path, when the flight distance is long and/or size of network is large, especially if the density of UAVs is high. To resolve this matter, the second proposed solution, referred to as Accelerated MAUP (AMAUP), uses a shortest weighted path algorithm considering directed graphs, and that is to determine the desired paths.

The reminder of this paper is organized as follows. Section II presents the related works. Section III discusses our network model and problem formulation. The MAUP solution is detailed in Section IV, whereas Section V presents AMAUP solution. Section VI presents the performance evaluation and results analysis. Finally, Section VII concludes the paper.

II. Related work

The UAV flight path optimization problem is considered in several a-priori works from different perspectives related to the targeted application. Each work aims at optimizing certain parameters such as the consumed energy, the flight duration, the traveled distance and the efficiency of task offloading. The design of UAV trajectory with MEC computation offloading scheduling, subject to the UAVs maximum speed and MEC nodes' computation capacity constraints, is discussed in [6]. However, the proposed solution is simplistic assuming a system with a single UAV per application that can be served from any potential MEC node, considering also the possibility of partitioning a task to smaller sub-tasks. Cheng et al [9] introduce another approach with different assumptions for the UAVs' trajectory optimization in cellular-aided UAVs networks. In the proposed solution, the UAV travels between the edge of three adjacent cells to provide data offloading services. The goal of the optimized trajectory is to maximize the sum rate of UAV served users, subject to the rate requirements for all users.

In [10], a UAV-enabled wireless-powered MEC system is studied and a power minimization problem is formulated by jointly optimizing the number of the offloading computation bits and the UAV flight path. A similar UAV-based MEC system is presented in [11], where the UAV trajectory was optimized under latency and UAVs energy budget constraints. In [12], an ant colony optimization algorithm is explored for the effective UAV path planning, considering obstacle-avoidance constraints. Luo et al [13] discussed an optimal trajectory planning strategy for a UAV-based inspection system of a large-scale photovoltaic farm using the Bezier curve and a particle swarm optimization algorithm.

An interference-aware path planning scheme for a network of cellular-connected UAVs, wherein each UAV aims at achieving a trade-off between maximizing energy efficiency and minimizing both wireless latency and the interference level, is proposed in [14]. A novel path planning algorithm for performing energy-efficient inspection using UAVs under stringent energy availability constraints is proposed in [15]. The optimal trajectory planning for UAV-assisted data collection in wireless sensor networks under the age of data constrains is elaborated in [16]. Work in [17], aims at optimizing the trajectory of a swarm of UAVs while avoiding the physical collisions and minimizing the mission's delay and the consumed energy. In [18], the energy-efficiency in UAV to Ground (U2G) communication is addressed via UAV's trajectory optimization.

Our proposal complements the current state of the art by introducing a unified flight planning and network resource optimization solution taking into account edge cloud resources at MEC platforms. The objective is to allocate the optimal UAV's trajectory assuring the desired UAV service for the entire flight path, while minimizing the total travel distance.

III. Network Model and Problem Formulation

A. Definition

Definition 1. Given a weighted graph $G = (V, E, W)$, a start node $S \in V$ and a destination node $D \in V$, P is the set of possible paths from S to D, with the weight of a path $p =< v_0, v_1, ...v_k >$ for $p \in P$, where $v_i \in V$ is the sum of the weights of its constituent edges:

$$\omega(p) = \sum_{i=1}^{k} \omega_{i-1,i}.$$

Let $W(P) = \{\omega(p), \forall p \in P\}$ denote the set of weights of paths in P. A shortest path from S to D is then defined as a path $p \in P$ with weight $\omega(p) = Minimum(W(p))$.

B. Network model

We consider a cellular network of stationary MEC nodes (i.e., co-located at base-stations) denoted as N. Let $\mathcal{U}$ be a set of UAVs where each $u \in \mathcal{U}$ is associated to a single pair (u_S, u_D) with $(u_S, u_D) \in (N \times N)$ and $u_S \neq u_D$. u_S denotes the MEC node in the source location of the UAV and u_D denotes the MEC node at the desired UAV destination. Each MEC node $i \in N$ is associated with a resource capacity $\mathcal{R}_i$, and each UAV $u \in \mathcal{U}$ introduce a resources demand r_u.

Let $G = (V, E, W)$ be a weighted graph, where the set of vertices V represents the MEC nodes, that is, $V = N$ and the set of the edges E denotes neighborhood relations between MEC nodes (i.e., adjacency). W represents the edge weights, where $\omega_{i,j} \in W$ denotes the weight of the edge $(i, j) \in E$, which is equal to the distance between their corresponding MEC nodes.

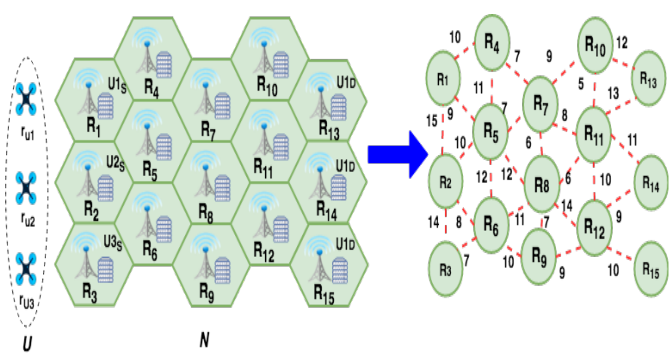


Fig. 1: Network model.

C. Problem formulation

In this paper, we propose two strategies for the UAV path planning to ensure the availability of MEC resources for computation offloading and control along the trajectory path, while minimizing the travel distance. We assume that the resources in a MEC node $i \in \mathcal{N}$, denoted $\mathcal{R}_i$, are decreased with r_u once a UAV $u \in \mathcal{U}$ selects that MEC node on its flight trajectory. The r_u resources are reserved for the UAV $u \in \mathcal{U}$ before starting the flight and are released once the flight is completed. We assume that the UAVs initiate their flights from a MEC node with sufficient resources. A summary of the notations used in this paper is presented in Table I.

TABLE I: Summary of Notations.

Notation	Description
$\mathcal{N}$	The set of MECs nodes.
$\mathcal{U}$	A set of UAVs in the network.
$\mathcal{D}(u)$	A function $\mathcal{D} : \mathcal{U} \longrightarrow \mathcal{N}$ that returns the destination of a UAV $u \in \mathcal{U}$.
$\mathcal{S}(u)$	A function $\mathcal{S} : \mathcal{U} \longrightarrow \mathcal{N}$ that returns the starting location of a UAV $u \in \mathcal{U}$.
$\eta(a)$	Set of neighbors of a vertex $a \in V$. Formally, $\eta(a)$ consists of a set of vertices, whereby there is an edge E between a and those vertices.
r_u	The amount of resources needed by UAV $u \in \mathcal{U}$.
$\mathcal{R}_i$	The resource capacity of MEC $i \in \mathcal{N}$.
$\mathcal{M}$	A big number ($\mathcal{M} \approx +\infty$).
$\lambda((i,j))$	A weighted function that returns the weight of an edge (i,j) in a weighted directed graph.
$G(V,E,W)$	A weighted graph with $V = \mathcal{N}$ and E is a set of edges. Formally, an edge $(i,j) \in E$ only if $i,j \in \mathcal{N}$ are physically neighbors. W denotes the weight of the graph G, such that $w_{i,j} \in W$ denotes the distance between two adjacent MEC nodes $i,j \in \mathcal{N}$.
$\mathcal{X}_{i,j}^u$	A Boolean variable that shows if a UAV $u \in \mathcal{U}$ has moved from a MEC $i \in \mathcal{N}$ to another MEC $j \in \mathcal{N}$. Formally, $\mathcal{X}_{i,j}^u = 1$, if UAV u moves from the MEC i to the MEC j. Otherwise, $\mathcal{X}_{i,j}^u = 0$

IV. MEC-Aware UAVs' Path planning

Hereafter, we describe our optimal solution for MEC-Aware UAVs Path planning (MAUP). MAUP models the MEC-Aware path planning process as an integer linear program, that aims to provide an optimal path in terms of MEC resource availability and the length of the traveled distance for the maximum number of UAVs. We define the following variables:

$\forall u \in \mathcal{U}, \forall i \in \mathcal{N}, \forall j \in \eta(i)$:

$$\mathcal{X}_{i,j}^u = \begin{cases} 1 & \text{If UAV } u \in \mathcal{U} \text{ passes from node } i \text{ to node } j \\ 0 & \text{Otherwise} \end{cases}$$

MAUP's linear program model is presented as follows:

$$\mathbf{min} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{N}} \sum_{j \in \eta(i)} \mathcal{X}_{i,j}^u \times \omega_{i,j}$$

s.t.

$\forall u \in \mathcal{U}$:

$$\sum_{j \in \eta(\mathcal{S}(u))} \mathcal{X}_{\mathcal{S}(u),j}^u = 1 \tag{1}$$

$\forall u \in \mathcal{U}$:

$$\sum_{j \in \eta(\mathcal{D}(u))} \mathcal{X}_{j,\mathcal{D}(u)}^u = 1 \tag{2}$$

$\forall u \in \mathcal{U}$:

$$\sum_{j \in \eta(\mathcal{D}(u))} \mathcal{X}_{\mathcal{D}(u),j}^u = 0 \tag{3}$$

$\forall u \in \mathcal{U}, \forall i \in \mathcal{N},$:

$$\sum_{j \in \eta(i)} \mathcal{X}_{i,j}^u \leq 1 \tag{4}$$

$\forall u \in \mathcal{U}, \forall i \in \mathcal{N} \setminus (\mathcal{S}(u) \cup \mathcal{D}(u))$:

$$\sum_{j \in \eta(i)} \mathcal{X}_{j,i}^u = \sum_{j \in \eta(i)} \mathcal{X}_{i,j}^u \tag{5}$$

$\forall i \in \mathcal{N}$:

$$\sum_{u \in \mathcal{U}} \sum_{j \in \eta(i)} \mathcal{X}_{j,i}^u \times r_u \leq \mathcal{R}_i - \sum_{u \in \mathcal{U} \wedge \mathcal{S}(u) = i} r_u \tag{6}$$

MAUP is modeled as a minimization problem, where the aim is to minimize the total traveled distance by UAVs. Constraint 1 ensures that each UAV in the network will move from its starting location to exactly one MEC node, while Constraint 2 ensures that each UAV will reach its destination coming from exactly one MEC node. Constraint 3 ensures that a UAV will stop when reaching its destination. Constraint 4 ensures that each UAV will move from a MEC node to exactly one other MEC node, while Constraint 5 assures that an allocated path to a UAV is not interrupted. Furthermore, the combination of Constraints 1, 2, 4 and 5 ensures that the selected flight path does not contain a cycle loop. Finally, Constraint 6 ensures that each MEC node satisfies the resource requirements of the set of transversing UAVs.

V. Accelerated MEC-Aware UAV Path Planning

Given its complexity, the proposed optimized solution elaborated in the previous section may introduce scalability concerns, especially when the flight distance is long and hence a high number of path possibilities and MEC nodes exist along the trajectory. A high number of UAVs may further contribute to increasing the complexity of the solution. In this section, we propose an accelerated algorithm for MAUP with polynomial run time complexity.

Accelerated MAUP (AMAUP) is based on the notion of shortest path between two vertices defined in the Subsection III-A. Indeed, after removing all MEC nodes with insufficient resources for a given UAV $u \in \mathcal{U}$ from the network represented by the graph $G = (V, E, W)$, a new directed weighted graph is calculated and a shortest path calculation algorithm is applied to determine the flight trajectory of that UAV.

A. Dijkstra Algorithm

Let $G(V, E, W)$ be a positive weighted graph. Dijkstra's algorithm is used for finding the shortest path between a source vertex $S \in V$ and a destination vertex $D \in V$. The algorithm works in three steps:

1) The algorithm starts by assigning a tentative distance value for each $v \in V$. This value is set to 0 for S and to infinity for all other nodes. Then, the algorithm keeps track of two sets, (i) the set of visited vertices VST initialized to the empty set, and (ii) a set of unvisited vertices $UVST$ initialized to V, that is $VST = \{\emptyset\}$ and $UVST = \{V \setminus S\}$.
2) Next, the algorithm selects an unvisited vertex with the smallest tentative distance as the current node CN. It then updates the tentative distance of its unvisited neighbors, i.e., $j \in \{neighbors(CN) \cap UVST\}$ to the minimum value considering the current tentative distance of j and the cost for reaching j from S via CN i.e., $min[tentative(j), tentative(CN) + w(cn, j)]$. In completing this step, the CN is marked as a visited vertex and removed from $UVST$.
3) Step 2 is repeated until the destination node D is marked as visited or $UVST$ becomes empty.

B. AMAUP description

In order to provide a comprehensive insight of AMAUP, Algorithm 1 and the detailed example depicted in Fig. 2 are used to elaborate the numerous execution steps. Fig. 2(a) depicts the input graph G, the set of UAVs and their source and destination locations as well as their resource requirements including the resource capacities of MEC nodes, which are represented by the numbers inside the vertices.

AMAUP starts by sorting the set of UAVs according to their resource demand (Algorithm 1: line 2). This will later serve to minimize the probability of network partitioning during the execution of the algorithm. Then, the input graph $G = (V, E, W)$ is converted to a directed weighted graph $dG = (dV, dE, dW)$ (Algorithm 1: lines 3 – 9) by transforming each edge $(i, j) \in E$ into two directed edges (i, j) and (j, i) in dE. The weight of the new directed edges is calculated using the weighted function $\lambda((i, j)) = w_{i,j}/\mathcal{R}_j$, where $w_{i,j}$ is the weight of the edge (i, j) in the graph G and $\mathcal{R}_j$ is the resource capacity of the MEC node represented by the vertex j. The resulted graph is illustrated in Fig. 2(b). The weighted function λ makes the MEC nodes with the higher resource capacity favored and accessible with low cost. This ensures load balancing between the MEC nodes, while minimizing the probability of network partitioning.

The next step is the calculation of flight paths for the sorted set of UAVs. Indeed, the algorithm passes iteratively on the sorted set of UAVs and performs the following actions for each UAV $u \in \mathcal{U}$:

1) Generate a sub-graph $\dot{dG}$ of dG by removing all the vertices that do not satisfy the resource demand of u (Algorithm 1: lines 11 – 32). In the example illustrated

---

Algorithm 1 Algorithm of AMAUP

Input:
  $G = (V, E, W)$: Network model graph.
  $\mathcal{U}$: The set of UAVs.
  r: The set of resources needed by UAVs.
  $\mathcal{R}$:The set of resources available at MEC nodes.
Output:
  $\mathcal{P}$: The set of paths.

1: $\mathcal{P} = \{\emptyset\}$;
2: $Sort(\mathcal{U})$;
  Convert G **to a directed weighted graph** dG
3: $dV = V$
4: **for all** $(i, j) \in E$ **do**
5:   $dE = dE \cup \{(i, j), (j, i)\}$
6:   $dW[i, j] = W[i, j] \div \mathcal{R}[j]$
7:   $dW[j, i] = W[i, j] \div \mathcal{R}[i]$
8: **end for**
9: $dG = (dV, dE, dW)$
10: **for all** $u \in \mathcal{U}$ **do**
  Construct a sub-graph: $\dot{dG} = (\dot{dV}, \dot{dE}, \dot{dW})$
11:   $\dot{dV} = dV$;
12:   $\dot{dE} = dE$;
13:   $\dot{dW} = dW$;
14:   **for all** $v \in \dot{dV}$ **do**
15:     **if** $(\mathcal{R}[v] < r[u])$ **then**
16:       $\dot{dV} = \dot{dV} \setminus v$;
17:       **for all** $(i, j) \in \dot{dE}$ **do**
18:         **if** $i == v$ **or** $j == v$ **then**
19:           $\dot{dE} = \dot{dE} \setminus (i, j)$;
20:           $\dot{dE} = \dot{dE} \setminus (j, i)$;
21:           $\dot{dW} = \dot{dW} \setminus \dot{dW}[i, j]$;
22:           $\dot{dW} = \dot{dW} \setminus \dot{dW}[j, i]$;
23:           **if** $i == v \;\&\; \eta(j) == \emptyset$ **then**
24:             $\dot{dV} = \dot{dV} \setminus j$;
25:           **else if** $\eta(i) == \emptyset$ **then**
26:             $\dot{dV} = \dot{dV} \setminus i$;
27:           **end if**
28:         **end if**
29:       **end for**
30:     **end if**
31:   **end for**
32:   $\dot{dG} = (\dot{dV}, \dot{dE}, \dot{dW})$;
  Calculate the shortest path:
33:   $\mathcal{P}[u] = Dijkstra(\dot{dG}, \mathcal{S}(u), \mathcal{D}(u))$
  Reserve resources:
34:   **for all** $v \in dV$ **do**
35:     **if** $v \in \mathcal{P}[u]$ **then**
36:       $\mathcal{R}[v] = \mathcal{R}[v] - r[u]$;
37:       **for all** $(i, j) \in dE$ **do**
38:         **if** $j == v$ **then**
39:           **if** $\mathcal{R}[v] == 0$ **then**
40:             $dW[i, j] = \mathcal{M}$
41:           **else**
42:             $dW[i, j] = W[i, j] \div \mathcal{R}[v]$
43:           **end if**
44:         **end if**
45:       **end for**
46:     **end if**
47:   **end for**
48: **end for**
49: **return** $\mathcal{P}$

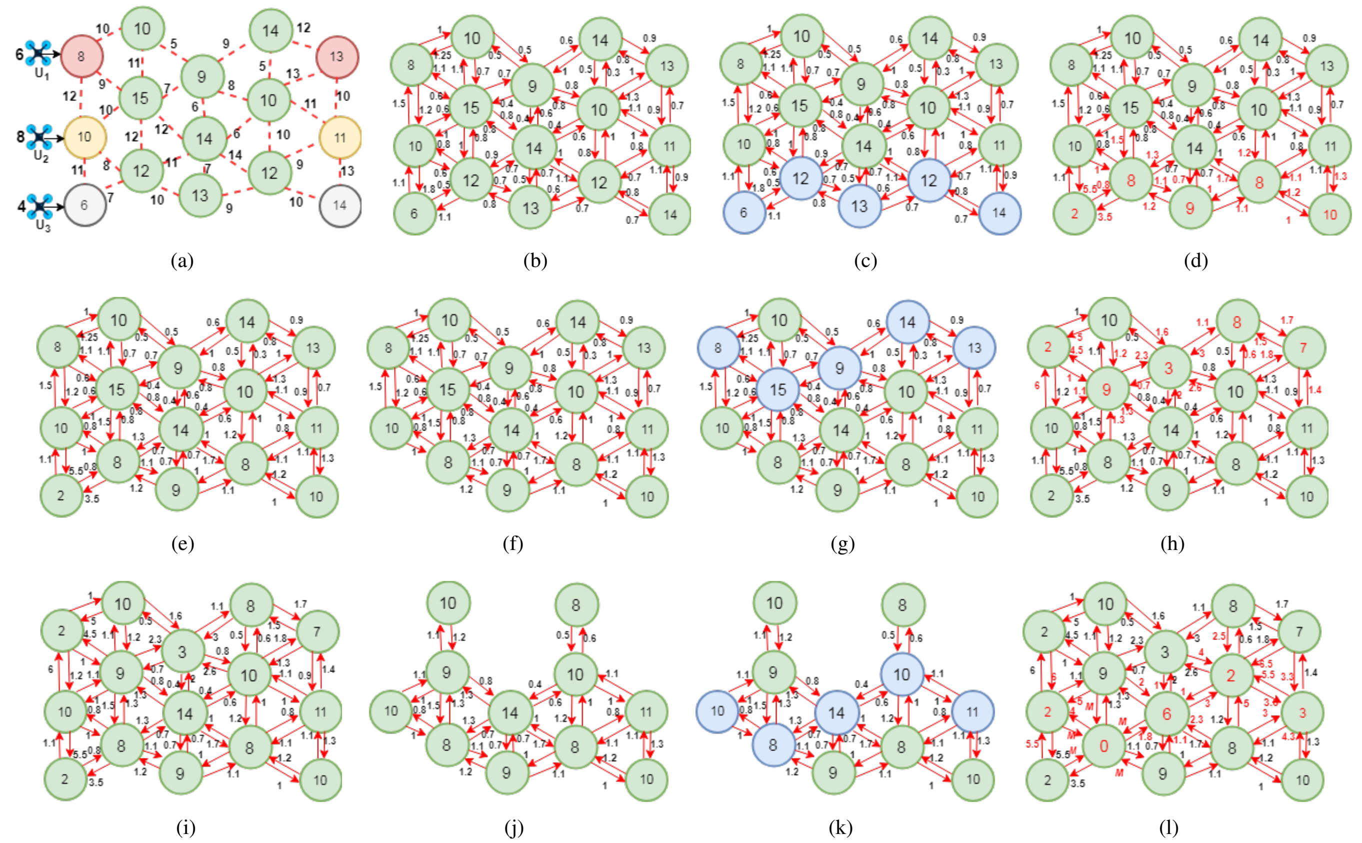


Fig. 2: Execution of the AMAUP solution.

in Fig. 2, none of the vertices are removed in the first iteration as all MEC nodes satisfy the resource demand of u_3. However, some vertices are removed later in the 2nd and 3rd iterations as depicted in Fig. 2(f) and Fig. 2(j), respectively.

2) AMAUP uses the Dijkstra algorithm to find the flight path to the destination with lowest cost (Algorithm 1: line 33). As stated before, the weighted function λ favours the selection of MEC nodes with high resource capacity and small distance from the source location. Fig. 2(c), Fig. 2(g) and Fig. 2(k) depict the selection of the flight paths for UAVs u_3, u_1 and u_2.
3) Finally, the algorithm updates dG by decreasing the amount of the available resources on the set of nodes that belong to the selected path, as well as the weights of the set of edges with their endpoints in the selected path (Algorithm 1: lines 34 – 43).

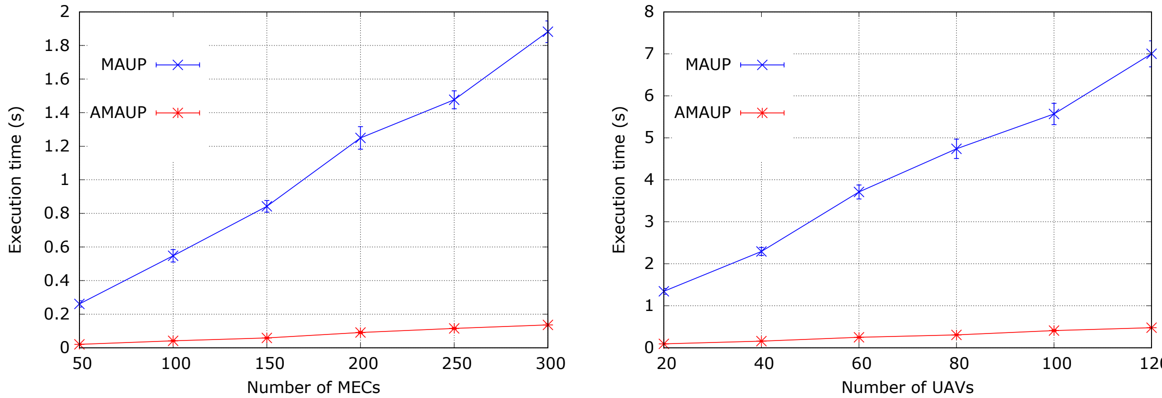
The use of the weighted function λ minimizes the probability of partitioning the sub-graph $\dot{dG}$ during the first step of the iterative paths calculation phase. However, this probability is not eliminated and $\dot{dG}$ may get partitioned. Hence, it may be impossible to find a path for a candidate UAV.

VI. Performance Evaluation

In this section, we present our simulation setup and discuss the obtained results. To the best of the authors' knowledge, there is no similar work in the literature that has the same solution objectives. For this reason, herein, we evaluate only our solutions, MAUP and AMAUP, without comparing them to other base-line approaches. The solutions are evaluated in terms of: $i)$ run time, which is defined as the time needed to execute each solution; $ii)$ the average distance traveled by each UAV. While the former metric shows the complexity of the proposed solution, the latter captures the cost in terms of energy consumption and travel time. The longer the distance that is traveled, the longer the travel time and hence the more energy is consumed.

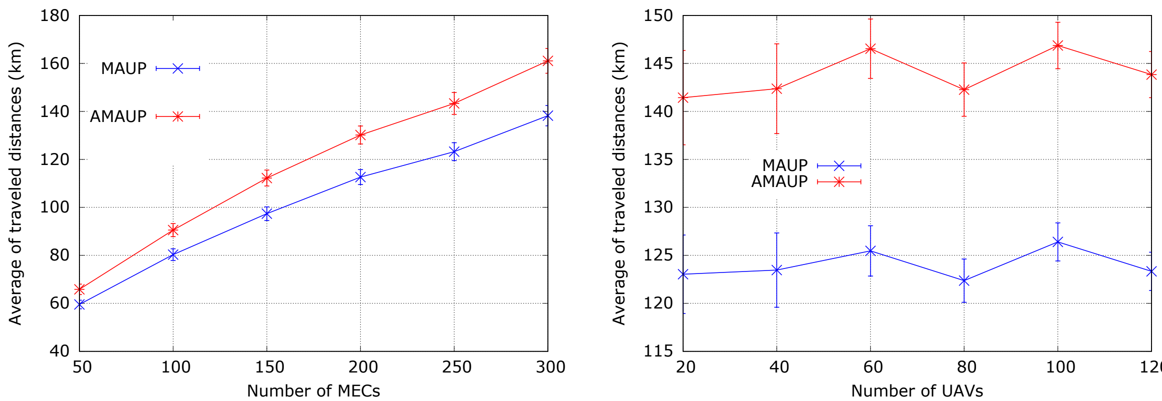
We evaluated the former metrics by varying the number of MEC nodes and UAVs. For each scenario, we run 50 repetitions, altering the UAVs starting and destination locations, the resource demand of UAVs and the resource capacities of MEC nodes. The MAUP solution was implemented using the GUROBI tool. Meanwhile, we developed a simulator using Python and the extended package related to graph theory called NetworkX for evaluating the AMAUP solution. All our experiments were conducted on a multi-core server with the configuration described in Table II.

In the simulation results, each plotted point represents the average of the 50 repetitions and the plots are presented with 95% confidence interval.



(a) Execution time. (b) Execution time.

Fig. 3: Run time complexity.



(a) Average of traveled distances. (b) Average of traveled distances.

Fig. 4: The average of traveled distances.

TABLE II: Hardware Configuration.

Type	**Configuration**
CPU	Intel(R) Xeon(R) CPU E5-2680 V3 @ 2.50GHZ
RAM	256 GB
OS	Ubuntu 16.04
Kernel	4.4.0-124-generic

A. Run time complexity

Fig. 3(a) and Fig. 3(b) show the run time complexity of MAUP and AMAUP as a function of the number of MEC nodes $|\mathcal{N}|$ and the number of UAVs $|\mathcal{U}|$, respectively. As depicted in these figures, increasing the number of MEC nodes or UAVs results in increased run time complexity for both MAUP and AMUP solutions. A higher number of UAVs has a bigger impact on the run time complexity, especially for the MAUP one. Overall, Fig. 3 shows that AMAUP outperforms MAUP irrespective of the number of MEC nodes and UAVs. From Fig. 3(a), the difference in the run time between MAUP and AMUP increases sharply and reaches more than 12 times higher. Also the gap in execution time between AMAUP and MAUP widens with the increase in UAVs number, where MAUP takes up to 17 times more than AMAUP, as illustrated in Fig. 3(b).

B. Traveled distance

Fig. 4(a) and Fig. 4(b) show the average traveled distance related to the set of UAVs as a function of the number of MEC nodes $|\mathcal{N}|$ and the number of UAVs $|\mathcal{U}|$, respectively. The results clearly show that MAUP outperforms AMAUP in term of the average traveled distance, with the performance advantage growing as the number of MEC nodes increases. As illustrated in Fig. 3(a), the gap between MAUP and AMAUP raises from 10% when the number of MEC nodes is 50 to 15% when the number of MEC nodes is up to 300. However, the performance advantage of MAUP on AMAUP is nearly stable when varying the number of UAVs. As depicted in Fig. 4(b), the difference between the MAUP and AMAUP is almost constant at 14%. AMAUP cannot select the optimal paths in terms of the length of the traveled distance as the algorithm alters the weight of vertices among the MEC nodes. This is accomplished using the weighted function λ to minimize the probability of network partitioning and to ensure load balancing between MEC nodes. However, the obtained results are encouraging given the trade-off between the run time complexity and the traveled distances.

VII. Conclusion

In this paper, we introduced the novel problem of edge cloud resource-aware flight planning for UAVs, proposing two solutions, namely MAUP and AMAUP. MAUP uses an integer linear programming model that minimizes the total length of the selected paths, while ensuring adequate resources availability in MEC nodes for each UAV. On the other hand, AMAUP uses the notion of shortest paths in a weighted graph to minimize the length of the selected paths. MAUP aims to find the optimal solution, meanwhile, AMAUP aims to reduce the run time complexity at the cost of a slight degradation in comparison with MAUP. The simulation results demonstrated the effectiveness of the proposed solutions in achieving their design goal. As future research, we intend to improve the proposed solutions by taking in consideration time windows for allocating edge resources to UAVs rather than the end-to-end strategy used in this paper. Furthermore, the capability of UAVs to connect to multiple cells at the same time will be explored, especially to solve the problem of network partitioning faced in the AMAUP solution.

Acknowledgment

This work is partially supported by the 6G Flagship with grant agreement No. 318927, and by the European Unions Horizon 2020 research and innovation program under the Primo-5G project with grant agreement No. 815191. It is also supported in part by the Academy of Finland under CSN project with grant No. 311654.

References

[1] N. Hossein Motlagh, T. Taleb, and O. Arouk, “Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives,” *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, Dec 2016.

[2] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “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, thirdquarter 2017.

[3] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, “Survey on multi-access edge computing for internet of things realization,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2961–2991, Fourthquarter 2018.

[4] O. Bekkouche, T. Taleb, and M. Bagaa, “Uavs traffic control based on multi-access edge computing,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 1–6.

[5] N. H. Motlagh, M. Bagaa, and T. Taleb, “Uav-based iot platform: A crowd surveillance use case,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, February 2017.

[6] X. Cao, J. Xu, and R. Zhang, “Mobile Edge Computing for Cellular-Connected UAV: Computation Offloading and Trajectory Optimization,” *ArXiv e-prints*, Mar. 2018.

[7] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, “Towards edge slicing: Vnf placement algorithms for a dynamic and realistic edge cloud environment,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

[8] (2018, May) Unmanned aircraft system (uas) traffic management (utm): Concept of operations. FAA. [Online]. Available: https://utm.arc.nasa.gov/docs/2018-UTM-ConOps-v1.0.pdf

[9] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, R. Yu, and V. C. M. Leung, “Uav trajectory optimization for data offloading at the edge of multiple cells,” *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2018.

[10] F. Zhou, Y. Wu, H. Sun, and Z. Chu, “Uav-enabled mobile edge computing: Offloading optimization and trajectory design,” in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[11] 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, March 2018.

[12] J. Chen, F. Ye, and T. Jiang, “Path planning under obstacle-avoidance constraints based on ant colony optimization algorithm,” in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, Oct 2017, pp. 1434–1438.

[13] X. Luo, X. Li, Q. Yang, F. Wu, D. Zhang, W. Yan, and Z. Xi, “Optimal path planning for uav based inspection system of large-scale photovoltaic farm,” in *2017 Chinese Automation Congress (CAC)*, Oct 2017, pp. 4495–4500.

[14] U. Challita, W. Saad, and C. Bettstetter, “Cellular-connected uavs over 5g: Deep reinforcement learning for interference management,” *CoRR*, vol. abs/1801.05500, 2018. [Online]. Available: http://arxiv.org/abs/1801.05500

[15] M. Monwar, O. Semiari, and W. Saad, “Optimized path planning for inspection by unmanned aerial vehicles swarm with energy constraints,” in *GLOBECOM 2018 - 2018 IEEE Global Communications Conference*, Dec 2018, pp. 1–6.

[16] J. Liu, X. Wang, B. Bai, and H. Dai, “Age-optimal trajectory planning for uav-assisted data collection,” in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2018, pp. 553–558.

[17] S. Ouahouah, J. Prados-Garzon, T. Taleb, and C. Benzaid, “Energy and delay aware physical collision avoidance in unmanned aerial vehicles,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 1–7.

[18] Y. Zeng and R. Zhang, “Energy-efficient uav communication with trajectory optimization,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, June 2017.