Collaborative Trajectory Optimization for Outage-aware Cellular-Enabled UAVs

Amirahmad Chapnevis*, İsmail Güvenç†, Laurent Njilla‡ and Eyuphan Bulut*
*Dept. of Comp. Science, Virginia Commonwealth University, Richmond, VA 23284
†Dept. of Elec. and Comp. Engineering, North Carolina State University, Raleigh, NC 27607
‡Air Force Research Laboratory, Rome, NY
Email: {chapnevisa, ebulut}@vcu.edu, iguvenc@ncsu.edu, laurent.njilla@us.af.mil

2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring) | 978-1-7281-8964-2/20/$31.00 ©2021 IEEE | DOI: 10.1109/VTC2021-Spring51267.2021.9448776

***Abstract*—Cellular-enabled unmanned aerial vehicles (UAVs) require almost continuous cellular network connectivity to fulfill their missions successfully. However, the area (e.g., rural) they fly over may have partial coverage, making the path planning of such UAV missions a challenging task. Recently a tolerable outage duration is taken into account for such UAVs, and the trajectory optimization under this outage duration is studied. However, these existing studies consider only a single UAV and focus on optimization of each UAV's own path separately even in multi-UAV scenarios. In this paper, we study the trajectory optimization problem for cellular-enabled UAVs by taking into account the collaboration among UAVs. That is, for a given set of UAVs, each with a mission to fly from a starting point to an ending point, we aim to optimize the total mission completion time for all UAVs such that none of them has a connection outage more than a threshold. We let UAVs collaborate and provide connectivity as relays to each other to solve their outage problem and shorten their trajectories. We first model and solve this problem using nonlinear programming after discretization of the problem. Since it takes longer to solve the problem with such an approach, we then provide a graph-based approximate solution that runs fast. Numerical results show that the proposed approximate solution provides close to optimal results and performs better than state-of-the-art solutions that consider each UAV separately without collaboration among UAVs.**

***Index Terms*—UAV, trajectory optimization, cellular network.**

I. Introduction

Unmanned aerial vehicles (UAVs) have recently been utilized in many different applications such as surveillance and communication [1]. In order to benefit from UAVs truly in practice, however, it is significant to make sure that UAVs have secure, reliable and low-latency communication links with the ground control stations for their command and control. However, current products in the market today rely on direct Line-of-Sight (LoS) communication with their pilots in the ground over nonlicensed spectrum. Thus, a new approach [2] that aims to control UAVs through cellular connection has been considered in order to enhance the performance of UAV based systems. In such a scenario with cellular-enabled UAVs [3], the ground base stations (GBSs) provide connection to UAVs so that they can communicate with their pilots on the ground as well as with other UAVs.

This work is supported in part by Air Force Research Lab (AFRL) and Commonwealth Cyber Initiative (CCI). DISTRIBUTION A. Approved for public release. Distribution unlimited. Case Number AFRL-2021-0120. Dated 20 Jan 2021.

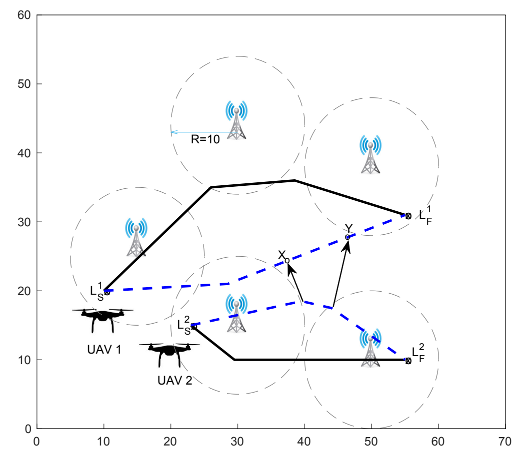

Fig. 1: An example scenario with 5 GBSs and 2 UAVs, where each UAV needs to travel from a starting location (e.g., $L_{\rm S}^1$) to a final destination point (e.g., $L_{\rm F}^1$). The UAV 1 can shorten its path by the help of UAV 2 as shown with dashed lines.

When the UAVs are used in rural areas, however, the cellular coverage that can provide the desired throughput between the UAVs and the GBSs may not be sufficient due to the smaller number of GBSs in the area. Thus, in some recent studies [4]–[7] a threshold on the outage (i.e., not having a cellular connection with required throughput) of these UAVs has been introduced and finding the best path for UAVs without exceeding this threshold has been studied. However, these studies optimize each UAV's path separately without considering their collaboration to maintain connectivity.

An example scenario is illustrated in Fig. 1, with five GBSs and two UAVs. The mission for each UAV is to fly from a start location to a final location without having more than a given cellular outage duration (which can be determined based on application requirements such as the frequency of control and command messages needed for the UAVs) continuously. The solid lines show the optimal trajectory of UAVs if they do not benefit from one another, while the dashed lines show the paths of UAVs when they help each other to maintain their connectivity at the desired level. Between points X and Y the first UAV connects to GBSs over the second UAV in a multi-hop fashion. Note that the second UAV changes its trajectory slightly in order to provide that support. Overall, the mission completion time for all UAVs decreases compared to the previous case with solid lines. Our goal in this paper is to consider such collaboration between UAVs in order to first let UAVs find a feasible path and then reduce their path lengths if possible as in the case of dashed lines.

The rest of the paper is organized as follows. We discuss the related work in Section II. In Section III, we provide the system model and provide the problem statement. As this problem is difficult to solve optimally, in Section IV, we present a graph-based approximate solution through several steps. In Section V, we provide numerical results regarding the performance of proposed solutions and a state-of-the-art solution that does not consider collaboration between UAVs. Finally, we conclude and discuss future work in Section VI.

II. Related Work

Cellular-enabled UAVs act as user equipments and are served by the GBSs to fulfill their mission. Thus, their connectivity is critical for completing their mission successfully. There are several studies performed recently looking at different issues for such cellular-enabled UAV networks including interference management [8], power control [9], and trajectory optimization [3], [10], [11] by considering several constraints such as privacy [12], and fading effects [6].

Due to the limited flight time of UAVs, the minimization of mission completion time has been one of the main problems studied. Moreover, it is also studied jointly with some other objectives. For example, in [13], the authors investigated the joint UAV trajectory and user scheduling to maximize the minimum downlink throughput of terrestrial users. UAV trajectories and transmit power are jointly optimized with the objective of maximizing the minimum average rate among all users. To overcome the non-convexity optimization, the authors consider coordinate descent and decrease the computation complexity with successive convex optimization techniques.

Recent studies [4]–[7] introduced the outage constraint into the path optimization problem and different solutions have been proposed. In [4] a dynamic programming-based solution is studied to obtain approximate paths within polynomial time. In [5], [7], a graph-based approach is considered and through the usage of shortest path algorithms approximate paths for the UAVs are found. In [6], the fading effect is also taken into account to provide realistic results. However, all these studies consider only a single UAV based optimization in this outage-aware cellular-enabled UAV path problem. As it is shown in Fig. 1, in missions with multiple UAVs this approach however misses the collaboration opportunity among UAVs and calculates each UAV's path separately based on the coverage by GBSs. In this paper, we extend these efforts and study this problem considering the collaboration of UAVs.

Note that the problem studied here is different than the literature that study multi-hop relaying [14]–[16]. These studies indeed aim to reduce the latency of data (that is sent from ground users or sensors to GBSs) transmission through multiple hops. On the other hand, we consider path optimization of UAVs in a collaborative manner such that they help each other for their cellular connectivity.

Notations	Description
$\mathcal{U}$, $\mathcal{G}$	The set of UAVs and GBSs, respectively.
n, k	Number of UAVs and GBSs, respectively.
L^u_{S}, L^u_{F}	Start and final location of UAV u, respectively.
$x_u(t)$, $y_u(t)$	Location of UAV u in timeslot t.
$c_u(t)$	Connectivity of UAV u at time t. It is equal to 1 if UAV u can communicate to a GBS directly or over another UAV at timeslot t; otherwise it is 0.
R_G	Max distance/range for a GBS-UAV link to maintain required SNR level.
R_U	Max distance/range for a UAV-UAV link to maintain required SNR level.
T_u	Flight duration time of UAV u
V_u	Maximum speed of UAV u
T_{Total}	Sum of flight durations of all UAVs
τ_{max}	Maximum continuous outage threshold for UAVs

TABLE I: Notations and their descriptions.

III. System Model

A. Assumptions

We assume that there are n UAVs, denoted by the set $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$. The mission of each UAV u is to fly from a start location, $L^u_{\mathrm{S}} = (x^u_{\mathrm{S}},\, y^u_{\mathrm{S}},\, z^u_{\mathrm{S}})$ to a final destination point, $L^u_{\mathrm{F}} = (x^u_{\mathrm{F}},\, y^u_{\mathrm{F}},\, z^u_{\mathrm{F}})$, where $z^u_{\mathrm{S}} = z^u_{\mathrm{F}} = H \ \forall u \in \mathcal{U}$. The UAVs will fly with a constant speed of V at a constant altitude of H meters (m) without having a continuous outage of $\tau_{\max}$ time units. The set of GBSs are denoted by $\mathcal{G} = \{g_1, g_2, \ldots, g_k\}$, with $|\mathcal{G}| = k$. The location of i^{th} GBS, g_i, is denoted as (x_i, y_i, z_i) and the altitude of all the GBSs is assumed to be the same (i.e., z_i =z_j=H_{G} $\forall i, j \in [1, k]$).

Similar to previous work [4]–[6], for simplicity, we assume that each communication link (GBS-UAV or UAV-UAV) is allocated with orthogonal spectrum, thus there is no interference between them. Each GBS and UAV is equipped with a single antenna with omni-directional unit gain and the communication channel is assumed to be dominated by the LoS link. Given the specifications of the GBSs, a range R_G is defined for minimum Signal-to-Noise-Ratio (SNR) needed for the communication between the UAV and the GBSs. This could be computed using $R = \sqrt{\frac{\gamma_0}{S_{\min}} - (H - H_{\mathrm{G}})^2}$ where $\gamma_0 = \frac{P\beta_0}{\sigma^2}$ denotes the reference SNR, with P as the transmission power of each GBS, σ^2 as the noise power at the UAV receiver, and β_0 as the channel power gain at the reference distance of 1 m. S_{min} is the minimum required SNR value for the communication between the UAVs and the GBSs in the application. We assume that the SNR at the receiver UAV shows the connectivity quality for the cellular-enabled UAV communication. For UAV-UAV links, the required range, denoted by R_U, is also computed similarly. Table I shows the notations used throughout the paper.

B. Problem Statement

In the proposed scenario, the objective is to let the UAVs complete their mission without having a cellular outage duration (i.e., no direct or multi-hop link (over other UAVs) to one of the GBSs) more than $\tau_{\max}$ time units and minimize the total mission time (i.e., defined by the last UAV that reaches its final location) and later the total traveling distance of all UAVs. As the UAVs will be moving, we denote the location of UAV u at time t with $u(t)$ = ($x_u(t)$, $y_u(t)$, H), and $0 \leq t \leq T^u_{\max}$, where $T^u_{\max}$ is the maximum possible flying time of the UAV u with a constant speed of V_u. We then define the nonlinear optimization problem as follows:

$$\min \quad (\max_{\forall u \in U} T_u)\lambda + T_{Total} \tag{1}$$

$$\text{s.t.} \quad (x_u(0), y_u(0)) = (x^u_{\mathrm{S}}, y^u_{\mathrm{S}}), \forall u \in \mathcal{U} \tag{2}$$

$$(x_u(T_u), y_u(T_u)) = (x^u_{\mathrm{F}}, y^u_{\mathrm{F}}), \forall u \in \mathcal{U} \tag{3}$$

$$\mathrm{dist}^{u(t+1)}_{u(t)} \leq V_u, \forall u \in \mathcal{U}, \forall t \in T \tag{4}$$

$$c_u(t) \;=\; \begin{cases} 1, & \text{if } \exists g_k \in \mathcal{G} \text{ s.t. } \mathrm{dist}^{g_k}_{u(t)} \leq R_G \\ 1, & \text{if } \exists v \in \mathcal{U} \text{ s.t. } \mathrm{dist}^{v(t)}_{u(t)} \leq R_U \;\& \\ & c_v(t) \;=\; 1 \\ 0, & \text{otherwise.} \end{cases}$$

$$\forall u \in \mathcal{U}, \forall t \in T \tag{5}$$

$$\sum_{l=t}^{t+\tau_{max}} c_u(l) \geq 1, \forall u \in \mathcal{U}, t \in T \tag{6}$$

$$T_u = \sum_{t=0}^{T} (\mathrm{dist}^{u(t+1)}_{u(t)} / V_u), \forall u \in \mathcal{U} \tag{7}$$

$$T_{Total} = \sum_{u=0}^{U} T_u, \forall u \in \mathcal{U} \tag{8}$$

where,

$$\mathrm{dist}^v_u = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2 + (z_u - z_v)^2}\ .$$

Here, in (1), we use scalarization method (by multiplying the first goal with a large constant λ) and aim to first minimize the overall mission time and then minimize the total travel time of all UAVs with this mission time. (2) and (3) set the first and final locations of UAVs, respectively. (4) makes sure the UAVs do not travel more than their speed between consecutive time points. (5) sets their connectivity to 1 if they are in the range of a GBS or another UAV which is connected. Here, $c_u(t)$ denotes the connectivity of UAV u at time t. (6) is used to make sure there is no outage more than τ_{max} duration, (7) calculates the total flight duration of UAV u and (8) computes the total flight time of all UAVs.

IV. Proposed Solution

While the problem can be solved with a nonlinear optimization solver using the model provided in previous section, it takes longer time to reach the solution. Thus, in this section, we provide an approximate solution using a graph-based approach. First, we provide a feasibility check to see if the UAVs can reach to their destinations under the given constraints and environment (e.g., GBS locations) while also considering the collaborations among UAVs. If it is possible, then we provide a graph-based approximate path calculation for all UAVs.

A. Graph-based Feasibility Check

We first create a graph $G_u = (V, E)$ for each UAV u with nodes representing GBS locations and u's start and end locations and with edges created between the nodes if a UAV can fly between these nodes directly without exceeding the outage duration. The weight of the edges are also set as the Euclidean distance. More formally,

$$\begin{aligned} V &= \mathcal{G} \cup \{L^u_{\mathrm{S}}, L^u_{\mathrm{F}}\} \\ E &= \{e_{i,j} \mid \forall i \in V, \forall j \neq i \in V \text{ s.t.} \\ &\quad \mathrm{dist}^j_i \leq 2R_G + \tau_{max} \;\&\; w_{ij} = \mathrm{dist}^j_i\} \end{aligned} \tag{9}$$

Using the obtained graph and running the BFS algorithm from the start location of each UAV, we then find the set of GBSs each UAV can travel to by itself without being in the outage area more than τ_{max} duration.

Let $\mathbf{R}$ denote a $|\mathcal{U}| \times |\mathcal{G}|$ matrix showing the reachability of UAVs to the GBS areas. Thus, we set

$$\mathbf{R}_{u,g} = \begin{cases} 1, & \text{if UAV u can fly to GBS range g with a} \\ & \text{desired connection (i.e., no outage} > \tau_{max} \text{).} \\ 0, & \text{otherwise.} \end{cases}$$

We then find the number of UAVs that can reach to each GBS g as follows:

$$A_g = \sum_{\forall u \in \mathcal{U}} \mathbf{R}_{u,g}.$$

Algorithm 1: Feasibility Check

$\mathbf{R} = |\mathcal{U}| \times |\mathcal{G}|$: Reachability matrix
for *each* $u \in \mathcal{U}$ **do**
  Form graph G_u as described in (9).
  Run BFS from source u
  for *each* $g \in \mathcal{G}$ *that is reachable from* u **do**
    $\mathbf{R}_{u,g} = 1$
  end
end
for *each* $g \in \mathcal{G}$ **do**
  Calculate A_g using (10) to find the total number of UAVs that can reach g
end
continue=true
while *continue* **do**
  for *each* $u \in \mathcal{U}$ **do**
    Update the graph G_u using (10).
    Run BFS from source L^u_{S} and update $\mathbf{R}$
  end
  if $\sum_{\forall g \in \mathcal{G}} A_g$ *did not increase* **then**
    continue=false
  end
end
for *each* $u \in \mathcal{U}$ **do**
  if L^u_{F} *is not reachable from* L^u_{S} **then**
    return false
  end
end
return true

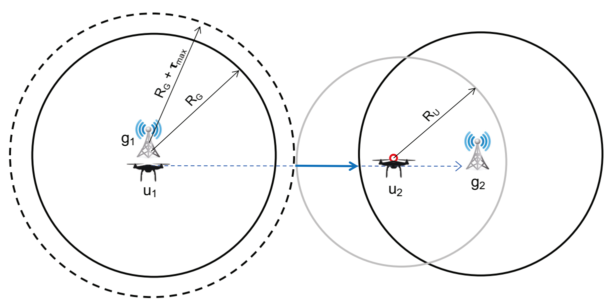

Fig. 2: The exact location of helping point in which u_2 helps u_1 to go to g_2's range from g_1's center.

Knowing A_g for each g, we then run the BFS algorithm again on each graph G_u and add new edges between each node pair if the following condition is satisfied.

$$\text{dist}_{g_i}^{g_j} \leq 2R_G + (A_i + A_j - 1)R_U + \tau_{max}. \quad (10)$$

This equation considers the best possible positioning of the UAVs (i.e., on the straight line connecting a GBS pair with R_U distance between them) in order to provide a coverage to the UAV of interest to travel from one GBS area to the other one. Note that running BFS should be repeated until there is no change in any of the A_g values. This is because as new edges are added to the graph, A_g can change for some $g \in \mathcal{G}$ thus new opportunities for adding edges can arise. Algorithm 1 shows the pseudo-code of this feasibility check. Once no more new edge is added, the algorithm checks if each UAV can reach its final destination and returns false if any of them cannot.

B. Graph-based Path Approximation

If the solution is feasible, in order to find the actual paths for all UAVs, we first add new vertices and edges to the graph, and then run Dijkstra's algorithm to find the shortest paths (considering w_{ij}s in (9)) from the starting point of each UAV to its final destination. Next, we go through these details.

1) Finding helping points: In feasibility check, if a UAV needs help of another UAV to pass from a GBS range to another, we first find these helping/relay points. Let h_i^j denote a helping point that u_i provides coverage to u_j for travel between two GBSs with a desired connection. Consider Fig. 2, where there is only one UAV that can reach to these GBS ranges independently (i.e., $A_g = 1$). Assume that u_1's shortest path from initial Dijkstra requires it to go to g_2's range from g_1's range (u_1 depends on u_2, or u_2 needs to help u_1). We then consider that u_2 should help u_1 at the location which is R_u away from the moment that u_1 fills its outage quota (i.e., border of dashed circle). In Fig. 2, this refers to the red point, which is $d = R_G + R_U + \tau_{max}$ away from g_1.

To calculate the exact location of this helping point, we first find the normal vector, $\hat{\mathbf{n}}(g_1, g_2)$, from g_1's center to g_2's center and multiply it with the distance of the helping point from g_1 and get the coordinates. That is, we get

$$\hat{\mathbf{n}}(u_1, u_2) = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} = \begin{bmatrix} \frac{x_{u_2} - x_{u_1}}{\text{dist}_{u_1}^{u_2}} \\ \frac{y_{u_2} - y_{u_1}}{\text{dist}_{u_1}^{u_2}} \end{bmatrix}.$$

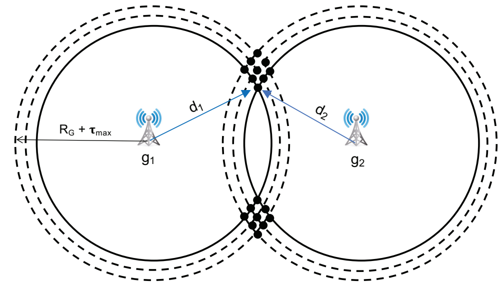

Fig. 3: Intersection of two GBS areas. Dark points are added to graph as new vertices.

Then, we get the point in which u_1 needs to be to help u_2 as:

$$h_i^j = (x_{u_1} + n_1 d, y_{u_1} + n_2 d).$$

u_1 needs to wait at this location at most $\text{dist}_{g_1}^{g_2} - 2R_G - \tau_{max}$ time units so that u_1 can safely reach g_2's range.

It is possible that the number of UAVs that can help u_1 can be more than what is actually needed. In such cases, we choose the UAVs with smaller shortest paths from initial Dijsktra run. Also, we currently do not consider circular dependency relation between UAVs and will consider this in future work.

2) Finding intersection points: In order to optimize the paths of UAVs further, we then add the intersection points between the GBS ranges. That is, we find the intersection points of the circles representing the boundaries of GBS service areas, then add them to the graph as new vertices. Note that such intersection points will always be in the service area and there will be no outage while traveling to these points from either of the GBS centers that form these intersections.

We also consider the outage area (i.e., between dashed and solid circles) and outage circles to find the intersection points. However, this time, in order to make sure outage threshold is not exceeded during travel of UAVs, and more smooth paths can be found, we first divide the outage area into several circular regions (i.e., adding more circles between solid and dashed circle) and after finding the intersection points of each of these circles between each other, we check if total the distance from that point to the GBS centers considered is less than or equal to $2R_G + \tau_{max}$. For example, in Fig. 3, we added one more circle between outage and service area circle and found 9 different intersection points in both upper and lower side. Then, for each of these intersection points we first check if $d_1 + d_2 \leq 2R_G + \tau_{max}$, then add to the graph if true.

Note that intersection points are just another set of points that a UAV can travel on its path. Thus, for those UAVs which get help from other UAVs on their shortest path calculation, we can also consider the range of helper UAVs as one of the circles that will intersect with other circles considered (e.g., circles of GBS service areas, and intermediate circles introduced in the outage area). For example in Fig. 2, one intersection that will result from this is the intersection of gray and dashed circles, which is happening at only one point for this specific example but could be two in other scenarios. These points are added to G_{u_1} or to the graph of UAV u_1.

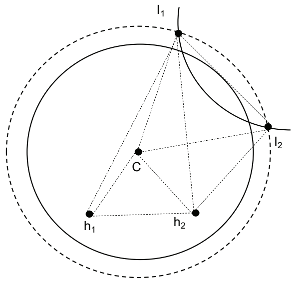

Fig. 4: The vertices and edges in the graph: I_1 and I_2 are intersections of circles, h_1 and h_2 are helping points of a UAV to another UAV, and C is the center of GBS's region. Edges are created between all vertices.

3) Adding edges between nodes on the graph: As new vertices are added to the graphs of UAVs, we also need to add new edges from these vertices. To this end, we first find the vertices originated from a service provider's location. These include the points of GBS locations and helping points of UAVs. Once these are found, for each of them, we find the set of all points included in their region and add an edge for each pair. Fig. 4 shows an example of this where an edge between all pair of nodes is added to the graph because they are in range of the same GBS service area.

4) Adding short-cut edges between nodes on the short-path: In order to optimize the path for UAVs, we also consider adding short-cut edges between the nodes in the UAV travel path graphs. This is because such paths may not be added following the aforementioned procedures as the distance of such edges can be longer than the specified condition checks. To this end, we first draw an edge between each pair of nodes on the current shortest path of a UAV and then perform a maximum outage check. To achieve this check, we find the intersection points of this new short-cut edge with the circles of GBSs and calculate their exact coordinates. Then, starting from one end point till the other one, we sort these intersection points based on their distance and check every consecutive two intersection points if they are in the range of different GBSs. If that is the case and the distance between these intersection points is more than τ_{max}, we do not add such a short-cut edge; otherwise an edge is added if all such consecutive intersection points pass this check. This process is summarized in Alg. 2.

Fig. 5 illustrates an example scenario for this process. There are six intersection points between the short-cut edge and GBS circles which are represented by set $M = \{M_1, M_2, M_3, M_4, M_5, M_6\}$. Checking consecutive intersections from different GBS service areas, we recognize that if the distance between M_1M_2 and M_3M_4 can be traveled without exceeding the maximum outage, then the short-cut edge is considered eligible. Running the Dijkstra after this step then provides a shorter path for the UAV compared to its current path that follows $< g_1, I_1, I_2, I_3, g_4 >$.

5) Path calculation and time synchronization: Once the graph of each UAV is finalized, we then run Dijkstra's algorithm to find shortest paths. Note that if a UAV is helping to another UAV, it needs to be present at a specific helping point

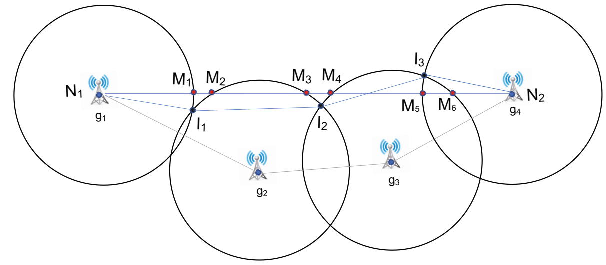

Fig. 5: Checking the feasibility of adding short-cut edge between two nodes, N_1 and N_2, on the current path of a UAV.

Algorithm 2: Adding Short-Cut Edges

for *each (N_1, N_2) pair on the current path of UAV* **do**
  Add a temporary edge E' between N_1 and N_2
  $M \leftarrow$ ordered set of intersection points between E' and GBSs.
  for *each consecutive points (M_i,M_{i+1}) in M* **do**
    if $dist^{M_i}_{Mi+1} \geq \tau_{max} \times V$ & M_i *and* M_{i+1} *are in range of different GBS service areas* **then**
      Remove E' as it is infeasible
    end
  end
end

when the other UAV needs it. Thus, for such helper UAVs we find their path to final destination points after visiting and waiting in these helping points. Once the paths of each UAV is found, we then need to synchronize their timing for their simultaneous travel. This requires addition of some additional waiting time to helper UAV's path if it arrives to the helping point early and needs to wait there until the depending UAV no longer needs it. Due to the space, we skip these details.

V. Numerical Results

In this section, simulation results to evaluate the performance of the proposed approximate solution are presented. Specifically, we used a map with 12 GBSs and 3 UAVs. We set the height of GBSs to 12.5m and the height of UAVs to 90m. Reference SNR (at 1 m) is set to 80 dB and S_{min} = 26.02 dB. Maximum speed of the UAVs is also set to 50 m/s. Under the given parameter settings, R_G and R_U can be derived as 10 units (of x-y axis shown in figures), which is equal to 250 m.

In Fig. 6, we first compare the optimal path of UAVs obtained via CPLEX using the nonlinear model in Section III and the paths obtained with approximate solution defined in Section IV. As the results show, the proposed approximate solution can obtain very close path to the optimal solution. Note that when there is no collaboration, UAV 2 cannot fly from its start location to final location when $\tau_{max} = 2.5$ time units. Thus, previous work [4]–[7] that do not consider collaboration among UAVs will not find a path for UAV 2 while it was possible with the help of UAV 1 and 3. Table II shows the path lengths for each UAV under each scenario.

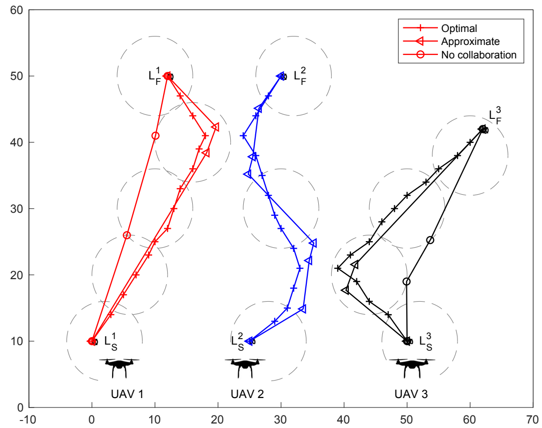

Fig. 6: Optimal and approximate UAV trajectories with collaboration of UAVs. When there is no collaboration UAV 2 cannot fly with given outage threshold ($\tau_{max} = 2.5$ units).

TABLE II: Path lengths of UAVs (shows in units of axis).

Method	**UAV 1**	**UAV 2**	**UAV 3**	**Total**	**Mission**
CPLEX	47.02	47.06	47.20	141.28	47.20
Heuristic	48.80	50.69	45.27	144.76	50.69
No collaboration	41.80	N/A	35.39	N/A	N/A
τ_{max}	**UAV 1**	**UAV 2**	**UAV 3**	**Total**	**Mission**
1	50.91	49.56	48.37	148.84	50.91
2.5	48.80	50.69	45.27	144.76	50.69
5	46.37	49.81	42.01	138.19	49.81

In Fig. 7, we also show the trajectories of UAVs obtained with approximate solution with different outage thresholds. As τ_{max} increases, the individual and total UAV path lengths and mission (defined by the longest UAV trip) durations decrease as expected. These results show the robustness of the proposed approximate solution which works under different settings.

When we compare the running time of the proposed approximate solution and CPLEX based optimal solution, for example for Fig. 6 results, we have 0.1s and ∼30 min, respectively. Thus, approximate solution provides much faster solution while providing close-to-optimal results.

VI. Conclusion

In this paper, we investigate the collaborative trajectory optimization problem for cellular-enabled UAVs under a given outage constraint. We consider a scenario in which each UAV aims to complete their mission by flying from a starting location to a final location and all UAVs collaborate with each other to maintain their connectivity with GBSs. We first formulate the problem using nonlinear optimization problem and in order to reduce the computation complexity, we then develop a graph-based approximate solution. Through numerical results, we show that the proposed approximate solution can provide close-to-optimal results and works for different scenarios. In our future work, we will extend the ideas here and also perform more simulations with extended scenarios.

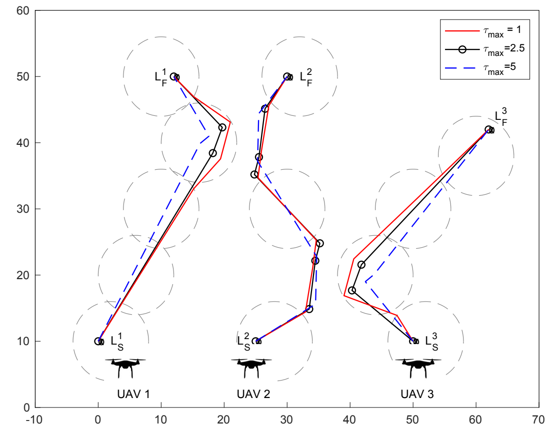

Fig. 7: Comparison of approximate trajectories with different connection outage thresholds.

References

[1] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Comm. Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[2] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: Potential, challenges, and promising technologies," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 120–127, 2018.

[3] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Trans. on Communications*, vol. 67, no. 3, pp. 2580–2604, 2018.

[4] E. Bulut and I. Güvenç, "Trajectory optimization for cellular-connected UAVs with disconnectivity constraint," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.

[5] S. Zhang and R. Zhang, "Trajectory design for cellular-connected UAV under outage duration constraint," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[6] Y.-J. Chen and D.-Y. Huang, "Trajectory Optimization for Cellular-Enabled UAV With Connectivity Outage Constraint," *IEEE Access*, vol. 8, pp. 29 205–29 218, 2020.

[7] O. Esrafilian, R. Gangula, and D. Gesbert, "3D-map assisted uav trajectory design under cellular connectivity constraints," in *IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[8] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected uavs: A deep reinforcement learning approach," *IEEE Trans. on Wireless Communications*, vol. 18, no. 4, pp. 2125–2140, 2019.

[9] W. Mei, Q. Wu, and R. Zhang, "Cellular-connected uav: Uplink association, power control and interference coordination," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5380–5393, 2019.

[10] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: Trajectory optimization under connectivity constraint," in *IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[11] H. Binol, E. Bulut, K. Akkaya, and I. Guvenc, "Time optimal multi-uav path planning for gathering its data from roadside units," in *88th IEEE Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.

[12] Y. Pan, S. Li, J. L. Chang, Y. Yan, S. Xu, Y. An, and T. Zhu, "An unmanned aerial vehicle navigation mechanism with preserving privacy," in *IEEE International Conf. on Communications (ICC)*, 2019, pp. 1–6.

[13] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-uav enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[14] U. Challita and W. Saad, "Network formation in the sky: Unmanned aerial vehicles for multi-hop wireless backhauling," in *IEEE Global Communications Conference (Globecom)*, 2017, pp. 1–6.

[15] S. Zhang, H. Zhang, Q. He, K. Bian, and L. Song, "Joint trajectory and power optimization for UAV relay networks," *IEEE Communications Letters*, vol. 22, no. 1, pp. 161–164, 2017.

[16] M. M. U. Chowdhury, S. J. Maeng, E. Bulut, and I. Güvenç, "3-d trajectory optimization in uav-assisted cellular networks considering antenna radiation pattern and backhaul constraint," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 3735–3750, 2020.