Communication-Aware UAV Path Planning

Communication-Aware UAV Path Planning

Afshin Mardani, Marcello Chiaberge, Paolo Giaccone
Dept. of Electronics and Telecommunications, Politecnico di Torino, Italy

***Abstract*—Autonomous air drones, known as Unmanned Aerial Vehicles (UAVs), often carry out missions with real-time video streaming. In this work, we address the off-line mission plan problem of computing the optimal path from an origin to a final destination in two-dimensional space in order to maximize the quality of the communication, given a cellular coverage, and thus to provide throughput guarantees to the video streaming. In addressing the problem, we consider the energy budget constraint, the presence of wind in the area and the path smoothing problem. We propose novel path planning algorithms that are shown to outperform classical approaches that are oblivious of communication network coverage.**

I. Introduction

Autonomous air drones are becoming more and more popular and typically require real-time video streaming for safety and surveillance reasons. In the case of long-distance flights, the drone can leverage the cellular network infrastructure to support the videostream communication. When planning an autonomous path between a set of distant waypoints, we advocate the adoption of path-planning algorithms which are aware not only of the obstacles (as in classical approaches) but also of the cellular coverage, in order to guarantee the stringent requirements of Quality of Service (QoS) in the communication. Thus, in our work we address the off-line path planning problem between two generic waypoints, given an energy budget and a bandwidth requirement in terms of either minimum throughput or average throughput. We consider the coverage map and possible effects of the wind that could affect the path trajectory. We also consider the problem of smoothing the path obtained by our path planning algorithms, in order to obtain straight flying trajectories and compensate for the spatial sampling of the adopted graphs.

Our main contributions are manyfold.

1) We propose two novel path planning algorithms, derived from the popular A* algorithm. MT-PP (Minimum Throughput Path Planning) is aimed at maximizing the minimum throughput along the path, in order to guarantee a minimum level of communication QoS. AT-PP (Average Throughput Path Planning) is instead aimed at maximizing the average throughput along the path, in order to guarantee an average level of communication QoS.
2) We show numerically that a communication-aware approach like ours can significantly outperform classical approaches (oblivious of cellular coverage), in terms of throughput, without exceeding the energy budget constraint.
3) We propose a novel path smoothing approach (IPS) that allows to decrease the computation time with respect to classical approaches.

The paper is organized as follows. In Sec. II, we describe the path planning problem. In Sec. III we discuss related works. In Sec. IV we propose our novel communication-aware path planning algorithms. In Sec. V we assess the achievable performance due to our proposed approach. Finally, we draw our conclusions in Sec. VI.

II. The path planning problem

We address the problem of generating the optimal trajectory from an initial waypoint to another waypoint in a two-dimensional space for an autonomous drone. In our innovative communication-aware approach, the quality of the video streaming application during the flight is maximized, taking into account the available energy budget and the expected wind conditions. Notably, the trajectory is computed off-line and fed to the drone internal controller, which will follow it compensating for the actual disturbances during the flight.

A. Flight area

The flight area is modelled according to a grid graph, as shown in Fig. 1. Indeed, the cellular coverage area is partitioned with a regular tessellation and each node is located at the center of each tile. The node is associated with a value of throughput given by the average value within the corresponding tile. The nodes corresponding to adjacent tiles are connected by an edge. In this work, we will consider 8-degree grid graphs, i.e. with 8 neighbors for each node but our approach can be extended to any other grid graph.

Formally, the flight area is described with an undirected grid graph $G = (V, E)$. Each node $i \in V$ is associated with physical position (x_i, y_i) and the throughput b_i experienced

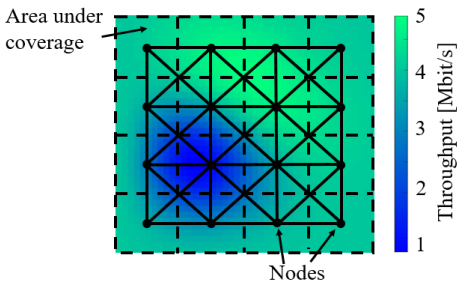

Fig. 1. Grid graph with degree 8 and the corresponding coverage map.

978-1-5386-4244-3/18/$31.00 © 2018 IEEE

by a drone when uploading the streaming data to the cellular network in the area around node i. An edge $(i,j) \in E$ connecting nodes i to j is associated with the physical distance d_{ij}.

For simplicity, we do not consider obstacles in the flight area, even if the methodology can be easily extended to this scenario by removing the edges and the nodes corresponding to obstacles.

B. Drone Mobility Model

We assume that the drone flies at constant *air speed* v_d with power consumption equal to P. The total energy available in the drone is E_0. The energy consumption due to traveling on an edge $(i,j) \in E$ is E_{ij}. The drone starts from node i_s (denoted as *source node*), corresponding to the starting waypoint, and arrives in node i_d (denoted as *destination node*), corresponding to the destination waypoint. A path comprising multiple waypoints is decomposed in a sequence of segments, for each of them the path planning algorithm runs.

Let $\mathcal{P}$ be the set of all possible loop-less paths connecting i_s to i_d and let $p \in \mathcal{P}$ be a generic path. The aim of the path planning is to find a path $p \in \mathcal{P}$ connecting i_s to i_d so that the communication performance is maximized, subject to the following energy budget:

$$\sum_{(i,j)\in p} E_{ij} \leq E_0 \tag{1}$$

C. Mobility Planning Between Two Nodes

Consider now a simplified problem in which the drone has to move from waypoint i to j. Let v_{dx} and v_{dy} be the drone *air speed*, along the two axes. Let v_{gx} and v_{gy} be the drone *ground speed*, along the two axes. Let t_{ij} be the time of flight from i to j. Assume that the wind is constant with speed v_{wx} along the x-axis and v_{wy} along the y-axis.

Now we can write the following system of equations:

$$v_{gx} = v_{dx} + v_{wx} \tag{2}$$
$$v_{gy} = v_{dy} + v_{wy} \tag{3}$$
$$x_j - x_i = v_{gx} \cdot t_{ij} \tag{4}$$
$$y_j - y_i = v_{gy} \cdot t_{ij} \tag{5}$$
$$v_{dx}^2 + v_{dy}^2 = v_d^2 \tag{6}$$

where (2) and (3) relate the effect of the wind to the actual ground speed; (4) and (5) relate the physical distance to travel to the ground speed; finally, (6) relates the air speed of the drone to the two components. By solving the above system, it is possible to compute the drone speed and its time of flight. Thus, the flight distance covered by the drone is $v_d \cdot t_{ij}$, and the energy consumption is:

$$E_{ij} = P \cdot t_{ij} \tag{7}$$

The ground distance from i to j is equal to the Euclidean distance between them. The total data transferred along the path i to j is $b_i \cdot t_{ij}$. All the introduced parameters with their descriptions are collected in Table I.

TABLE I
VARIABLE DEFINITIONS.

Parameter	Description	Unit
b_i	Cellular network throughput at node i	bit/s
d_{ij}	Physical distance	m
E_{ij}	Energy consumed between two nodes	J
E_0	UAV total energy	J
t_{ij}	Flight time	s
P	Power consumption	W
v_d	UAV air speed	m/s
v_g	UAV ground speed	m/s
v_w	Wind speed	m/s

III. Related Work

There are many path planning algorithms that have been developed in recent years [1]. In [2] the authors, inspired by the A* search algorithm and Dubins paths, present a path planning method to find the shortest, most flyable and safest path for fixed-wing UAVs with obstacle avoidance. [3] proposes any-angle path-planning algorithms, which are variants of the heuristic path-planning algorithm A*. Their algorithms find the shortest paths by propagating information along the grid edges (like A*, to be fast) without constraining the resulting paths to these edges. In [4], the authors propose a path planning method to generate paths for aircrafts in a 2-D space to avoid conflict, non-flying zones, and risk areas. Notably, none of these works consider network throughput in their path planning algorithms.

Path optimization for UAV communication systems has recently been investigated for different setups. In [5], a 3-D modeling of aerial wireless coverage is used in communication-aware path planning of UAVs for the aerial surveillance of long linear infrastructures. In [6], the UAV's flying direction is optimized for uplink communications. In [7] and [8], the authors optimize the movement of the UAVs to ameliorate the network connection of an ad-hoc network assisted by UAVs. However, none of the above works has integrated the power consumption of the drone.

Moreover, many path optimization problems have been studied, but not specifically for communication purposes. For example, in [9] the path is optimized considering the energy consumption of UAVs and not the communication performance. In [10], the task of reaching a specific goal from a set of possible goals is studied. The path is optimized in order to minimize the required energy, which depends on unknown disturbances, like the wind. Also in this case, the communication is not considered.

In some works, energy-efficient UAV communications are discussed, but without considering the effects of wind. In [11], the authors study energy-efficient designs for UAV communication, where a UAV is employed to communicate with the ground station. The energy efficiency is maximized via trajectory optimization. In [12], the problem of off-line path planning for UAVs is addressed, considering different mission objectives, as safety, fuel efficiency, collisions and possible criteria for communication. Nevertheless, all of the above

mentioned works neglect the wind effects on the power consumption.

There is not a unique approach to compensate for wind disturbances in path planning for UAVs. For instance, [13]–[15] compensate for the wind in the real-time control loop design of the UAVs to keep the position error of the flight path as small as possible with respect to the desired path. However, we compute the optimal path offline, taking into account the *expected* wind. The drone internal flight controller will compensate in real time for the actual wind conditions experienced while following the path computed according to the expected wind.

IV. Near-Optimal Communication-aware Path Planning

Our proposed approaches are based on the classical A* algorithm for path planning, adopted in many contexts as robotics and video games, and approximate the optimal path. A* is aimed at finding the shortest path (or a good approximation of it) with a much smaller computation time than canonical Dijkstra's algorithm, thanks to a smaller search subspace. Indeed, A* uses a cost function which is obtained by an heuristic cost function obtained by summing a function g, which represents the exact cost from the source to the current node (as in Dijkstra's algorithm), and a user-provided heuristic function h, which estimates the distance from the current node to the destination. Optimality and computation time depends on the chosen heuristic function. In particular, if h function gives exact distances to the destination (the estimated distance is equal to the distance on the grid), the algorithm only scans nodes on the shortest path from source to destination.

A. Path Planning Algorithms

Now, we introduce our path planning algorithms, which are variants of the A* algorithm.

1) AT-PP Algorithm: This algorithm maximizes the average throughput along the path, as:

$$\max_{p\in\mathcal{P}} \sum_{(i,j)\in p} \frac{b_i}{|p|} \tag{8}$$

subject to the energy budget. The pseudocode of the algorithm is provided in Algorithm 1, which neglects degenerate cases for the sake of readability. As in usual Dijkstra and A*, a node can be in three different states: (i) "visited" whenever the path cost to reach the node is already fixed and minimized, (ii) "frontier" whenever the node is one of the candidates for the minimum path cost, (iii) "unvisited" whenever the node has not been visited. The connected region of visited nodes keeps increasing until one node in the frontier reaches the destination, obtaining the best path. The cost function shown in ln. 39 is peculiar of our application: it combines the inverse of the average throughput ("path_bw") up to some node and the estimated distance to the final destination D. Consequently, the resulting path is optimizing both throughput and distance, weighted by factor β which is optimized numerically later.

The available energy budget E_0 is considered by pruning the visit whenever the consumed energy is above E_0 (see ln. 23). Notably, it may happen that no path is found since not compatible with E_0 (see ln. 40).

2) MT-PP Algorithm: This algorithm maximizes the minimum throughput along the path, as follows:

$$\max_{p\in\mathcal{P}} \min_{(i,j)\in p} b_i \tag{9}$$

The algorithm is iterative and works as follows. Assume an initial value of threshold t_h, which could be set equal to the average throughput achievable in the area. At each iteration, the algorithm finds the best path by considering only the nodes with a throughput $\geq t_h$. The path selection is based on a variant of AT-PP in which the throughput at some visited node is computed as the minimum along the path (differently from ln. 30 of AT-PP). To maximize the minimum throughput, at the end of the path search procedure t_h is increased. If no path is found with a minimum throughput $\geq t_h$ (maybe because of energy constraint), then t_h is decreased. To optimize the sequence of chosen values of t_h, we adopt a dichotomic search, which stops whenever a given precision on the minimum throughput along the path is achieved.

Algorithm 1 Average-Throughput Path Planning (AT-PP)

1: **function** AT-PP($\mathcal{N}, \{b_v\}_{v\in\mathcal{N}}, S, D, P, v_d, E_0, \beta$)
2: **for each** vertex $v \in \mathcal{N}$ **do** ▷ Initialization, for each vertex
3: path_bw[v]=-1 ▷ Throughput from S to v
4: acc_path_bw[v]=-1 ▷ Cumulative throughput from S to v
5: parent[v]=-1 ▷ Parent of node v
6: ground_path_distance[v]=∞ ▷ Ground distance from S to v
7: flight_path_distance[v]=∞ ▷ Flight distance from S to v
8: path_hops[v]=-1 ▷ Number of nodes from S to v
9: path_time[v]=-1 ▷ Flight time from S to v
10: path_energy[v]=∞ ▷ Energy from S to v
11: path_cost[v]=∞ ▷ Cost based on A*
12: path_bw[S]=acc_path_bw[S]=b_S ▷ Setting the values for S
13: parent[S]=S
14: ground_path_distance[S]=flight_path_distance[S]=path_hops[S]=0
15: path_time[S]=path_energy[S]=0
16: path_cost[S]=GroundDistance(S, D)
17: $\mathcal{U} = \mathcal{N} \setminus \{S\}$ ▷ Unvisited nodes
18: $\mathcal{F} = \{S\}$ ▷ Frontier nodes
19: $\mathcal{V} = \emptyset$ ▷ Visited nodes
20: **while** $\mathcal{F}$ is not empty **do** ▷ Visit all the nodes in the frontier
21: $u = \arg\min_{v\in\mathcal{F}} \{\text{path\_cost}[v]\}$ ▷ Find the min cost node in $\mathcal{F}$
22: move u from $\mathcal{F}$ to $\mathcal{V}$
23: **if** path_energy[u]> E_0 **then** ▷ Check the required energy to reach u
24: **continue** ▷ If greater than E_0, consider a new node in $\mathcal{F}$
25: **if** $u = D$ **then** ▷ Check if arrived to destination
26: **return** ▷ End. Return the whole state, from line 4 to 11
27: **for each** neighbor $v \notin \mathcal{V}$ of u **do** ▷ Check all the neighbors of u that are in $\mathcal{U}$ or $\mathcal{F}$
28: **if** $v \in \mathcal{U}$ **then**
29: move v from $\mathcal{U}$ to $\mathcal{F}$ ▷ Move v to the frontier, if is not there
30: bw=(acc_path_bw[u]+b_v)/(path_hops[u]+1) ▷ Average throughput along the path
31: path_bw[v]=bw
32: acc_path_bw[v]=acc_path_bw[u]+b_v
33: parent[v]=u
34: ground_path_distance[v]=ground_path_distance[u]+ GroundDistance(u, v)
35: flight_path_distance[v]=flight_path_distance[u]+ FlightDistance(u, v, v_w, w_{dir})
36: path_hops[v]=path_hops[u]+1
37: path_time[v]=path_time[u]+FlightDistance(u, v, v_w, w_{dir})/v_d
38: path_energy[v]=path_energy[u]+$P \times t_{uv}$
39: path_cost[v]=(flight_path_distance[u]+FlightDistance (u, v, v_w, w_{dir}))+FlightDistance(v, D, v_w, w_{dir})+β/path_bw[v] ▷ Main cost function
40: **return** error - destination unreachable

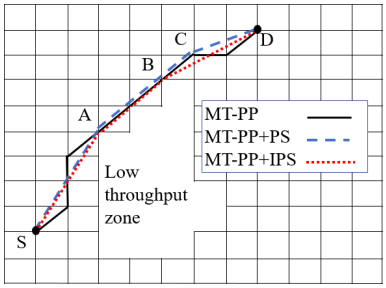

Fig. 2. An example showing the difference between our smoothing method A*IPS with respect to A*PS.

B. Path Smoothing

The paths resulting from AT-PP and MT-PP performing on grid graphs are constrained to be along the edges of the grid graph, thus are longer than shortest path on the free 2-D space. This observation leads us to smooth the paths, by a post-smoothing process increasing the runtime. We consider the A* post-smoothing algorithm (A*PS) proposed in [16]. A*PS usually finds a shorter path than A* on grid graphs. We propose a novel improved post smoothing process (A*IPS), according to which the resulting paths are in some cases shorter than A*PS. A*IPS works similarly than A*PS, with the difference that, when the A*PS is checking the line-of-sight (LOS) from the current node to the successor of its successor during the graph visit, A*IPS, in parallel, is also checking the LOS from the successor of its successor to the destination node. As long as the LOS is found to the destination, the whole process terminates and the shortest path is obtained. As an example in Fig. 2, the A*PS from the starting point A must check the LOS from point A towards the destination point, node by node. Since A*PS does not have LOS to the nodes between point C and end point from point A, it terminates in point C and then starts again from C to the destination. However, A*IPS, from the starting point A, is simultaneously checking the LOS towards the destination and the LOS from each following node to the destination. Therefore, our method finds the LOS from point B to the destination and terminates its process. Note that LOS is defined in terms of throughput values; two points are considered in LOS if all the closest nodes in the direct path among them is above the current value of throughput.

In Sec. V-C we will show that the resulting path of A*IPS is shorter or equal to the A*PS, with a possible reduction in computation time.

V. Performance Evaluation

A. Scenarios

Our algorithms were tested on two coverage maps, reported in Fig. 3, on a 4 km×4 km area. In the first coverage map (left of Fig. 3) the throughput shows a peak in the center and then decreases linearly to a minimum value, then it is constant with a higher value than the minimum in the whole border. This allows to create a discontinuity in the coverage

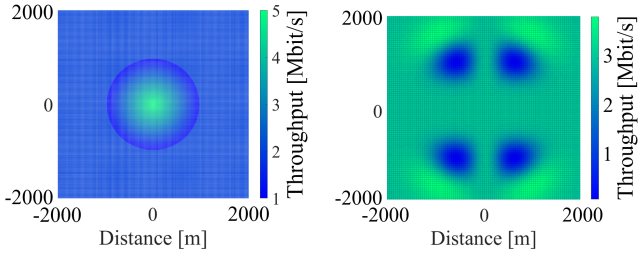

Fig. 3. Cone map (left) and valleys map (right).

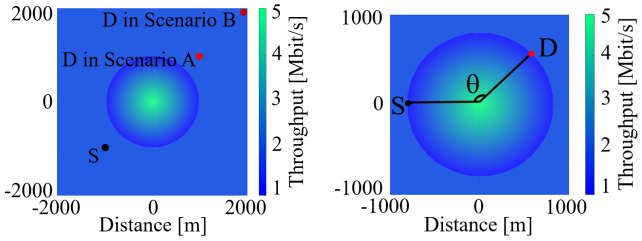

Fig. 4. Scenario A, B (left) and scenario C (right). "S" stands for source position. "D" stands for destination position.

that may impair the performance of greedy approaches (as the one considered in our work). Thus the cone map can be considered a simple worst-case coverage scenario. The second coverage map (right of Fig. 3) is denoted "valleys" and the coverage is now continuous with 4 valleys of low throughput.

The drone flies from a source position to a destination position. Combining the choice of different source-destination pairs with the coverage map, we defined the following scenarios:

- Scenario A (Fig. 4): This is symmetric case with the source in (-1000,-1000) m and the destination in (1000,1000) m, on the cone map.
- Scenario B (Fig. 4): This is an asymmetric case with the source in (-1000,-1000) m and the destination in (2000,2000) m, on the cone map.
- Scenario C (Fig. 4): The source and the destination are placed at θ degree from the peak of throughput on the cone map. This allows to investigate how a path is deviated towards the area of high throughput.
- Scenario D (Fig. 6): The source is in (-1000,-1500) m and the destination is in (1000,1300) m, on the valleys map. In this scenario, two low throughput regions are between the direct path from the source to the destination. This allows to show how the resulting path is deviated from low throughput areas.

When the wind is present, the wind can be either "head wind" (opposite direction than the direct path from the source to the destination) or "tail wind" (same direction than the direct path from the source to the destination).

B. Methodology

We investigate the behavior of our algorithms, both in dealing with limited energy on-board and the presence of wind, compared with the Shortest Path (SP) algorithm that computes

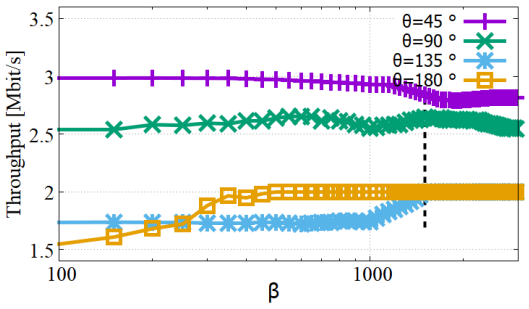

Fig. 5. Throughput in function of β under scenario C.

TABLE II
AVERAGE THROUGHPUT COMPARISON.

Scenario	Algorithm	Average Throughput [Mbit/s]		
		no wind	head wind 2 m/s	tail wind 2 m/s
D	MT-PP + IPS	3.3	2.6	3.5
	AT-PP + IPS	2.7	2.3	2.8
	SP	2.2	2.2	2.2
C $\theta = 135°$	AT-PP + IPS	2.8	1.5	2.8
	MT-PP + IPS	1.5	1.5	1.5
	SP	1.5	1.5	1.5
A	MT-PP + IPS	2.0	2.0	2.0
	AT-PP + IPS	2.5	2.5	2.5
	SP	2.5	2.5	2.5
B	MT-PP + IPS	2.0	2.0	2.0
	AT-PP + IPS	2.3	2.3	2.3
	SP	2.3	2.3	2.3

the direct path from the source to the destination position. SP has been chosen since it is representative of all the state-of-the-art algorithms presented in Sec. III, which are oblivious of the coverage map.

In our simulations we consider 8-degree graphs, as shown in Fig. 1, on a 101×101 grid with 40 meters of minimum distance between two nodes. Note that in the case of obstacles, the graph can be modified by removing the edges and the nodes that cannot be reached by the drone, and thus our algorithms can also take into account the presence of obstacles. For simplicity, we do not consider this case in this work.

Our algorithms are implemented in MATLAB and executed on a 2.67 GHz Core i7 PC with 8 GByte of RAM running Windows 10. Running times were evaluated using `tic` and `tac` MATLAB commands.

C. Numerical evaluation

First of all we optimize choice of the coefficient β utilized in cost function of our algorithms. Here we considered scenario C, and four destinations in $\theta = 45, 90, 135$, and 180 degrees are selected. Fig. 5 plots the throughput in function of β. We set $\beta = 1500$ (shown with dashed line in Fig. 5) to obtain a high throughput value in most of the cases.

Tables II and III report the average throughput and minimum throughput along the path, respectively, for all the considered scenarios and in presence of wind. Both MT-PP and AT-PP run with IPS as path smoothing procedure.

TABLE III
MINIMUM THROUGHPUT COMPARISON.

Scenario	Algorithm	Minimum Throughput [Mbit/s]		
		no wind	head wind 2 m/s	tail wind 2 m/s
D	MT-PP + IPS	3.0	1.0	3.1
	AT-PP + IPS	1.8	1.0	2.2
	SP	0.3	0.3	0.3
C $\theta = 135°$	AT-PP + IPS	0.1	0.1	0.1
	MT-PP + IPS	0.1	0.1	0.1
	SP	0.1	0.1	0.1
A,B	MT-PP + IPS	2.0	2.0	2.0
	AT-PP + IPS	0.1	0.1	0.1
	SP	0.1	0.1	0.1

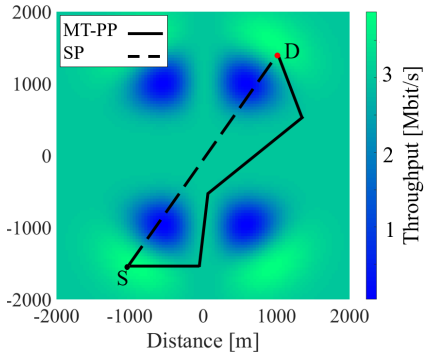

Fig. 6. Paths computed by SP and MT-PP under scenario D.

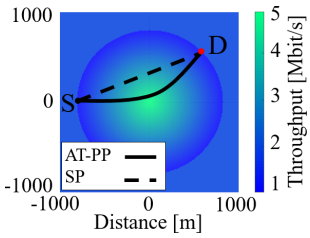

Fig. 7. Paths computed by SP and AT-PP algorithm under scenario C ($\theta = 135°$).

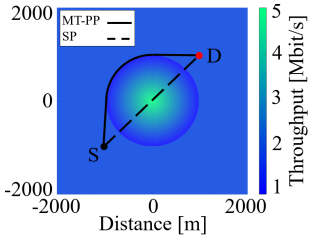

Fig. 8. Paths computed by SP and MT-PP algorithm under scenario A.

Consider for now the performance without wind. Scenario D, shown in Fig. 6, is validating the expected performance of MT-PP. The figure shows the paths obtained by MT-PP and SP. The average throughput achieved by MT-PP and AT-PP is larger than SP (50% larger for MT-PP) whereas the minimum throughput is much higher than SP (9 times larger for MT-PP.)

Fig. 7 shows the path found by AT-PP under scenario C,

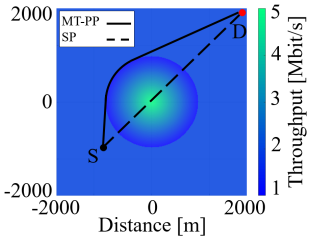

Fig. 9. Paths computed by SP and MT-PP algorithm under scenario B.

TABLE IV
COMPUTATION TIME AND PATH LENGTH COMPARISON.

Algorithm	Scenario	Run time [s]	Path length [m]
MT-PP	A	1.18	3507
	B	2.91	4922
	C ($\theta = 135$)	0.05	1535
	D	1.52	4715
MT-PP + PS	A	1.21	3325
	B	3.01	4655
	C ($\theta = 135$)	0.07	1418
	D	1.55	4450
MT-PP + IPS	A	1.21	3325
	B	2.94	4610
	C ($\theta = 135$)	0.05	1418
	D	1.57	4435

with $\theta = 135°$, validating the behaviour of the algorithm. Specifically the resulting path tends to pass close to the vertex of the cone with maximum throughput. This path is the optimal path considering that the average throughput is maximized. Thus, the average throughput of the path is increased up to 1.3 Mbit/s with respect to SP.

Moreover, in scenarios A and B, shown in Figs. 8 and 9, the minimum throughput achieved by the MT-PP algorithm outperforms the SP by about 20 times.

Consider now the effect of wind. In presence of the head wind, MT-PP and AT-PP gain less than the tail wind or when the wind is absent. This is due to the restricted energy budget to fly into the high throughput regions and compensate for the wind simultaneously. In scenario A and B, the path throughput of MT-PP is not affected by the wind in any cases, since MT-PP tends to avoid the low throughput regions so far as allowed by the energy constraint. In scenario C and D, in case of head wind, the MT-PP and AT-PP generated paths are closer to SP to be able to compensate for the wind effect and consequently the throughput of the resulting paths are lower than the no wind or tail wind cases.

We investigate the effect of the path smoothing algorithm in Table IV, where we compare the computation time and the corresponding path length for MT-PP alone (i.e., without PS), MT-PP with standard PS and MT-PP with our proposed IPS. By construction, MT-PP without path smoothing takes the lowest computation time, and adding standard PS takes 3.4% additional time, in the worst case (scenario B). Instead, adding IPS takes in the worst case (scenario B) up to 1.0% additional time. In terms of path length, under scenario A, PS and IPS reduce the path length by 5.1% whereas under scenario B, they reduce the path lengths by 5.4% and 6.3%, respectively. Thus, IPS is always preferred to PS in terms of both computation time and path length.

VI. CONCLUSIONS

In this paper, we investigated the path planning problem for an autonomous drone in order to optimize the quality of the video streamed by the drone. We considered the cellular coverage map, the energy budget of the drone and the possible presence of wind. We proposed two algorithms to maximize either the average throughput or the worst-case throughput along the path. We also proposed a novel path smoothing procedure outperforming the classical one in both terms of computation time and path length. Through a numerical evaluation of different scenarios, we showed that our communication-aware approach allows to improve the throughput with respect to classical approaches that are completely oblivious of the cellular coverage maps, with a beneficial effect on video streaming applications.

ACKNOWLEDGEMENTS

This work has been supported by PIC4SER (PoliTO Interdepartmental Centre for Service Robotics) laboratory at Politecnico di Torino.

REFERENCES

[1] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of UAV 3D path planning," in *WCICA*, IEEE, 2014.
[2] X. Song and S. Hu, "2D path planning with Dubins-path-based A* algorithm for a fixed-wing UAV," in *ICCSSE*, IEEE, 2017.
[3] A. Nash and S. Koenig, "Any-angle path planning," *AI Magazine*, 2013.
[4] J. Zeng, X. Zhang, and X. Guan, "Path planning for general aircrafts under complex scenarios using an improved NSGA-II algorithm," *Journal of Computational Information Systems*, 2013.
[5] H. Sharma and P. Balamuralidhar, "A communication approach for aerial surveillance of long linear infrastructures in non-urban terrain," in *NCC*, IEEE, 2015.
[6] F. Jiang and A. L. Swindlehurst, "Optimization of UAV heading for the ground-to-air uplink," *Journal on Selected Areas in Communications*, 2012.
[7] Z. Han, A. L. Swindlehurst, and K. R. Liu, "Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *Transactions on Vehicular Technology*, 2009.
[8] S. Kim, H. Oh, J. Suk, and A. Tsourdos, "Coordinated trajectory planning for efficient communication relay using multiple UAVs," *Control Engineering Practice*, 2014.
[9] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of UAVs," in *ICARSC*, IEEE, 2015.
[10] N. Bezzo, K. Mohta, C. Nowzari, I. Lee, V. Kumar, and G. Pappas, "Online planning for energy-efficient and disturbance-aware UAV operations," in *IROS*, IEEE, 2016.
[11] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *Transactions on Wireless Communications*, 2017.
[12] E. I. Grøtli and T. A. Johansen, "Path planning for UAVs under communication constraints using SPLAT! and MILP," *Journal of Intelligent & Robotic Systems*, 2012.
[13] K. Wu, B. Fan, and X. Zhang, "Trajectory following control of UAVs with wind disturbance," in *CCC*, IEEE, 2017.
[14] A. Rucco, A. P. Aguiar, F. L. Pereira, and J. B. de Sousa, "A predictive path-following approach for fixed-wing unmanned aerial vehicles in presence of wind disturbances," in *Robot*, Springer, 2016.
[15] S. U. Ali, M. Z. Shah, R. Samar, and A. Waseem, "Wind estimation for lateral path following of UAVs using higher order sliding mode," in *ICISE*, IEEE, 2016.
[16] C. Thorpe and L. Matthies, "Path relaxation: Path planning for a mobile robot," in *OCEANS*, IEEE, 1984.