II. Materials and methods

In this section, we dive into the problem and gradually progress to its solution through interrelated problems and different approaches.

A. Optimal route problem

We call "optimal flight plan" a route that is the shortest circuit possible, but still, guarantees that at the end of the cycle, the vision based system has captured the entirety of an assigned subregion. In practice, this study determines three properties that our flight plan should have:

1) The subregion is completely covered.
2) The flight plan is a cycle.
3) The flight plan is the shortest possible.

Problem II.1. *The "Coverage problem" is satisfied if after completing a flight plan, all the taken pictures, combined together, form a flawless view of the entire assigned subregion.*

First and foremost, we need to calculate the scope of the drone's camera. This information is fundamental to measure the maximal distance that the drone should travel to take another picture.

Definition II.1.1. *The "*True altitude*" is the altitude of an object relative to the* Mean Sea Level (MSL).

Definition II.1.2. *The "*Absolute altitude*" is the altitude of an object relative to the* Above Ground Level (AGL).

Definition II.1.3. *The $\Psi \equiv$ "Ground Sampling Distance (GSD)" [1] is the distance between the centres of two consecutive pixels, measured on the ground.*

Different formulas to compute the Ψ take into account the angle of the camera compared to the perpendicular of the ground and the topology of the ground [1]. Nonetheless, in this study, we adopt the following assumptions:

- The speed of the drone is constant.
- The absolute altitude is constant. Otherwise, if the elevation of the ground height changes (and drones maintain constant True altitude), Ψ would not be constant.
- The sensor width equals the sensor height.
- The camera is always perpendicular to the ground.

At that point we calculate Ψ as follows:

$$\Psi = \frac{100 \cdot (Sw \cdot H)}{Fr \cdot imgW} \quad [centimetres/pixel]$$

Where:

- $Sw \equiv$ Sensor witdh of the camera [millimeters]
- $H \equiv$ Flight height [meters]
- $Fr \equiv$ Focal length of the camera [millimeters]
- $imgW \equiv$ Width of the image [pixels]

Definition II.1.4. *We call "*scope of the camera*" the width Θ of a single image footprint on the ground.*

$$\Theta = \frac{\Psi \cdot imgW}{100} \quad [meters]$$

Therefore, in our case, a drone should travel a maximum of Θ m before taking another picture. Henceforth, this study adopts Θ as a fixed value.

1) Solution to problem II.1: To entirely cover the region assigned to a drone, we build a virtual grid of coordinates inside a polygon representation of the region. The distance between each coordinate of the grid (the unit) equals Θ. In practice, it means that if a drone takes a picture when flies over each point in the grid, then all the pictures combined, form a complete aerial image of the region.

Problem II.2. *The "Optimal route problem" states that the flight plan must be the shortest cycle and every point of the grid is visited.*

Definition II.2.1. *A graph $G = (V, E)$ is "induced" by a set $S \subset \mathbb{R}^2$ if $V = S, E = \{(i, j) \mid i, j \in S, |i - j| = 1\}$ (edges connect vertices at distance 1). [2]*

Definition II.2.2. *A "*grid graph*" is a finite graph induced by a subset of vertices of $\mathbb{Z}^2$($\equiv$ infinite square integer lattice) [2].*

Definition II.2.3. *A grid graph is "*solid*" if there are no holes in it.*

Definition II.2.4. *A "Hamiltonian cycle" is a cycle in a graph passing through each vertex exactly once. [3]*

In consideration of the above definitions, this study could focus on finding a Hamiltonian cycle of a solid grid graph, this type of problem is called "Grid Hamiltonicity" [2] [4]. Nonetheless, our algorithm needs to apply a third constraint by which the cycle must be the shortest.

Problem II.3. *Given a list of cities and their distances, the "*Travelling salesman problem - TSP*" asks to determine the shortest route that visits each city and returns to the origin city.*

Theorem 1. *Problem II.2 is a "Travelling salesman problem."*

Proof: We can reduce our problem to an instance of TSP. Firstly, we build a polygonal solid grid graph over the region to monitor. The distance between two adjacent points is Θ. Compared to the problem of "Grid Hamiltonicity", the drone can move from a point to each other point of the grid. Formally, our grid graph is a "complete graph" G; thus, every pair of distinct vertices is connected. The reason behind this choice is that we cannot advance any assumption concerning the most beneficial path that the drone should follow, even if it appears to be improbable that the drone goes directly to a distant vertex. Since every vertex G identify the coordinates of the points to be visited, we can assume that each vertex represents a city that the salesman needs to visit. Finally, the solution to such a TSP is also a solution to our problem. ∎

III. Results and Discussion

In this section of the article, we analyse the two proposed solutions from a theoretical perspective (correctness and characteristics) and a practical point of view (applicability and performance).

A. Optimal route problem

As previously mentioned, SANET solves Problem II.1 by first reducing it to an instance of the Travelling Salesman Problem. For efficiency and reliability reasons, the solution to the TSP is obtained through an open-source implementation called "Or-tools" developed by Google. In light of these circumstances, this study does not dwell on Google's resolution of the TSP, neither on other implementations of it. However, in the following paragraphs, we analyse the optimality and the correctness of the approach, relative to the initial problem: improving fight planning.

1) Optimality: From a theoretical perspective, our approach solves the primary problem optimally. Indeed, it provides a flight plan thanks to which drones can monitor the broadest accessible area in the most brief possible time. Nevertheless, the optimality of the solution strictly depends on the type of implementation, because there are exact and heuristic methods to solve TSP. These last are the most ordinary ones since they solve the problem more quickly by sacrificing optimality and precision. As we prove in the following paragraph, this is the best compromise in our context.

2) Complexity: The complexity of the solution depends on two factors. Most importantly, we should examine the complexity of the reduction. We implement a linear-time algorithm to determine the number of vertices $num(P) \equiv \omega$ in a region [5]. In the aftermath, our algorithm produces the edges by defining the connections between vertices. Since the graph is complete, there are $\frac{\omega(\omega-1)}{2}$ edges. The complexity of the reduction is, accordingly, $O\left(\frac{\omega(\omega-1)}{2}\right)$. In a second time, we have to consider the complexity of the solution to TSP. If we use a reference exact dynamic programming approach such as the "Bellman–Held–Karp algorithm" [6], the worst-case time complexity is $O(2^n n^2)$ (where $n \equiv$ number of cities) [7]. The computation for 30 nodes would require almost 1 trillion steps; hence it is unquestionably impractical. Conversely, heuristic methods lead to quadratic time complexity $O(n^2)$, such as in the case of the "Lin–Kernighan heuristic" [8]. In this case, the complexity of the entire proposed solution is $O(\omega^2)$. Therefore, a heuristic choice is the best compromise.

IV. State-of-the-Art

In this section, we explore other flight planning solutions and position our work by demonstrating the novelty of the research and its deficiencies. Most of works in this scientific field focus on flight planning of UAVs - Unmanned aerial vehicles, both for military purposes and civil purposes. In the military context, this work [9] presents an algorithm that generates a stealthy path trough enemy radars (which in our case can be identified as restricted areas, thus by absent coordinates in the grid graph). This work [10] instead appears more comparable to ours because it focuses on autonomous path solving in a cooperative setting. Besides, while in our work, we maximise the covered area, their algorithm maximises, straightforwardly, the objective function representing information gain through sensing systems. Oppositely to our assumptions, other works such as [11] take into account the complexity of real 3D environments. Since the angle of rotation represents a factor that determines energy consumption, certain researches such as [12] additionally considers those constraints to establish a path smoothness cost, consequently a more realistic cost function.

V. Conclusion

In this article, we focused on the enhancement of cooperative systems of drones. More concretely, we focused on the flight planning feature of a cooperative system. We converged on an alternative routing strategy which does not rely on complex optimization problems, but still maximize the information captured through the camera. As observed in the previous section, this work suffers from deficiencies related to the absence of additional realistic environmental and technical constraints. A future perspective would certainly enrich this work by adopting additional types of sensing equipment and more authentic flight constraints.

References

[1] Beatriz Felipe-García, David Hernández-López, and José Luis Lerma. Analysis of the ground sample distance on large photogrammetric surveys. *Applied Geomatics*, 4(4):231–244, 2012.

[2] Esther M Arkin, Sándor P Fekete, Kamrul Islam, Henk Meijer, Joseph SB Mitchell, Yurai Núñez-Rodríguez, Valentin Polishchuk, David Rappaport, and Henry Xiao. Not being (super) thin or solid is hard: A study of grid hamiltonicity. *Computational Geometry: Theory and Applications*, 42(6-7):582–605, 2009.

[3] Takanori Akiyama, Takao Nishizeki, and Nobuji Saito. Np-completeness of the hamiltonian cycle problem for bipartite graphs. *Journal of Information processing*, 3(2):73–76, 1980.

[4] Erik D Demaine and Mikhail Rudoy. Hamiltonicity is hard in thin or polygonal grid graphs, but easy in thin polygonal grid graphs. *arXiv preprint arXiv:1706.10046*, 2017.

[5] Hiroki Yanagisawa. A simple algorithm for lattice point counting in rational polygons. 01 2005.

[6] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.

[7] Charles Hutchinson, Jonathan Pyo, Luke Zhang, and Jieli Zhou. Cmu traveling salesman problem. 2016.

[8] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[9] Scott Bortoff. Path planning for uavs. volume 1, pages 364 – 368 vol.1, 10 2000.

[10] John Tisdale, ZuWhan Kim, and J Karl Hedrick. Autonomous uav path planning and estimation. *IEEE Robotics & Automation Magazine*, 16(2):35–42, 2009.

[11] Vincent Roberge, Mohammed Tarbouchi, and Gilles Labonté. Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning. *IEEE Transactions on Industrial Informatics*, 9(1):132–141, 2012.

[12] Zhangjie Fu, Jingnan Yu, Guowu Xie, Yiming Chen, and Yuanhang Mao. A heuristic evolutionary algorithm of uav path planning. *Wireless Communications and Mobile Computing*, 2018, 2018.