3D Real-Time Energy Efficient Path Planning for a Fleet of Fixed-Wing UAVs

# 3D Real-Time Energy Efficient Path Planning for a Fleet of Fixed-Wing UAVs

Giuseppe Aiello
Politecnico di Torino
10129 Torino TO, Italy
Email: s263610@studenti.polito.it

Kimon P. Valavanis
Ritchie School of
Engineering and Computer Science
University of Denver
Denver, CO 80210 – USA
Email: kimon.valavanis@du.edu

Alessandro Rizzo
Department of
Electronics and Telecommunications
Politecnico di Torino
10129 Torino TO, Italy
Email: alessandro.rizzo@polito.it

***Abstract*—UAV path planning requires finding an optimal (or sub-optimal) collision free path in a cluttered environment, while taking into account geometric, physical and temporal constraints, eventually allowing UAVs to perform their tasks despite several uncertainty sources. This paper reviews the current state-of-the-art in path planning, and subsequently introduces a novel node-based algorithm based on the called EEA*. EEA* is based on the A* Search algorithm and aims at mitigating some of its key limitations. The proposed EEA* deals with 3D environments, it provides robustness quickly converging to the solution, it is energy efficient and it is real-time implementable and executable. Along with the proposed EEA*, a local path planner is developed to cope with unknown dynamic threats in the environment. Applicability and effectiveness is first demonstrated via simulated experiments using a fixed-wing UAV that operates in different mountain-like 3D environments in the presence of several unknown dynamic obstacles. Then, the algorithm is applied in a multi-agent setting with three UAVs that are commanded to follow their respective paths in a safe way. The energy efficiency of the EEA* algorithm has also been tested and compared with the conventional A* algorithm.**

## I. Introduction

This paper focuses on fixed-wing UAVs and deals with real-time robust path planning in 3D cluttered environments, including dynamic collision avoidance, also considering physical constraints and limitations of the UAV itself. The problem is challenging since from the the optimization theory point of view, finding a 3D complete path is NP-hard.

A novel node-based path planning algorithm with real-time capabilities, named EEA*, is presented. It is suitable to navigate through 3D cluttered environments, and features a local path planner that copes with unknown dynamic threats within the operational space. Moreover, the energy efficiency of the EEA* algorithm is tested and compared with the conventional A* algorithm. It comes out that the EEA* reduces the work needed to follow the route in $86.7\%$ of the time and of the $3.43\%$ on average. Furthermore, also the average power dissipated has been indirectly decreased in $86.7\%$ of the cases of a $0.95\%$.

The rest of the paper is organized as follows. Section II offers a detailed literature review looking at the state-of-the-art, advantages and disadvantages as well as implementation challenges and what limits efficient implementation of existing path planning techniques. Section III states the problem and derives the proposed EEA* algorithm, while Section IV discusses implementation details. Section V concludes the paper.

## II. Literature Review

The UAV path planning problem is cast as an optimization problem that returns an optimal solution among all possible ones. According to [1], path planning algorithms can be divided in five categories: Sampling based algorithms, Node based optimal algorithms, Mathematical model based algorithms, Bio-inspired algorithms, Multi-fusion based algorithms. This classification is followed below to review related work.

*1) Sampling based algorithms:* In [3] and [14] two versions of the same algorithm have been used: RRT and RRT*. In the first one, the RRT has been combined with biased sampling and greedy extension of nodes with an added continuous curvature path smoothing satisfying non-holonomic constraints. The algorithm is very simple and fast enough to be executed online. In the second one, the RRT* has been used along with nadir- and oblique- camera views for close proximity as well as large-scale 3D mapping applications. With the support of state-of-the-art 3D reconstruction software, the recorded inspection data have been post-processed and dense and high-quality point clouds and triangular meshes have been derived. In [15] a hybrid approach has been employed for near-optimal solution in a 2D environment. A third dimension is managed in an online manner with a fuzzy controller working simultaneously with a Lazy Theta* algorithm. The best path may be found to avoid obstacles by changing the UAV's altitude online using data from the sensors and topographical database.

*2) Node based optimal algorithms:* In [4] a Fuzzy Virtual Forces (FVF) algorithm is used. To solve the local minima problem encountered in in VF algorithms, the combination of threats with an adjacent matrix has been integrated in the algorithm. An adaptive proportion coefficient based on Bayesian belief network and fuzzy logic reasoning has been added, which can be adapted to environment changes. In [22] a modified A* algorithm is proposed, combined with the Global Navigation Satellite System (GNSS) error distribution obtained by

TABLE I
SUMMARY OF UAV PATH PLANNING REVIEW. COLUMNS REPRESENT REFERENCED WORK AND YEAR PUBLISHED (REF), THE UAV TYPE UNDER CONTROL (UAV), EMPLOYED ALGORITHMS (ALG), ENVIRONMENT, WHETHER 2D (2D) AND/OR 3D (3D), REAL-IMPLEMENTATION (REAL-TIME), CHALLENGES NOT SATISFIED IN THE RESEARCH (LIMITATIONS).

| REF | UAV TYPE | ALGORITHM | LIMITS |
|---|---|---|---|
| [2] (2007) | Generic | VGA | 3 |
| [3] (2008) | Helicopter | RRT | 3,4 |
| [4] (2010) | Generic | FVF | 1 |
| [5] (2010 | Generic | ACO & DE | 2 |
| [6] (2011) | Generic | HVFA | 1,3 |
| [7] (2012) | Generic | MILP | 1,2 |
| [8] (2012) | Quadrotor | Flatness-based | 1,4 |
| [9] (2013) | Fixed-Wing | BLP | 1 |
| [10] (2013) | Fixed-Wing | GA, PSO | 3,4 |
| [11] (2013) | Quadrotor | PRM & A* | 3,4 |
| [12] (2013) | Quadrotor | HS | 1,2,3 |
| [13] (2014) | Generic | GA & PSO | 4 |
| [14] (2015) | Fixed-Wing, Quadrotor | RRT* | 3 |
| [15] (2015) | Generic | BF Lazy Theta* | 3 |
| [16] (2016) | Quadrotor | RRT | 3 |
| [17] (2016) | Quadrotor | Receding Horizon | 1 |
| [18] (2016) | Quadrotor | GWO | 1,2,3 |
| [19] (2016) | Fixed-Wing | HGA | 1,2,3 |
| [20] (2017) | Generic | MOPP & GA | 1,3,4 |
| [21] (2018) | Quadrotor | Q-Learning | 1,2,3,4 |
| [22] (2018) | Quadrotor | Modified A* | 2,3 |
| [23] (2019) | Hexarotor | Receding Horizon | 3 |
| [24] (2019) | Quadrotor | A*, Dijkstra | 1,3 |
| [25] (2019) | Generic | EA | 3 |
| [26] (2019) | Hexarotor | A* & S-PSO | 4 |
| [27] (2020) | Quadrotor | ACO, GLS, LKH | 1,2,3 |
| [28] (2020) | Generic | RLGWO | 2,4 |

implementing the GNSS, whereas in [25] researchers have proposed an Evolutionary Optimization Algorithm based on improved t-distribution to deal with high computational complexity.

*3) Mathematical model based algorithms:* In [7] mixed-integer programming for control (MILP) is used to successfully show the ability to plan paths that achieve a desired communication topology, while minimizing fuel consumption and avoiding collision and no-fly zones, also satisfying altitude constraints with respect to the terrain. In [8] a trajectory planning/replanning flatness technique is proposed that can be implemented in real-time applications deploying a simplified model. But using the simplified models in trajectory planning increased uncertainties and mismatch with the system. In [9] a BLP based approach is proposed that includes a BLP model and a solution algorithm embedded with heuristic strategies, while in [37] a BLP-based real-time path planner is introduced to generate reference way-points and control inputs at variable planning time intervals. Performance variations are addressed and smooth flight paths are adapted only when necessary. In [16] a given bounded volume is explored in a receding horizon manner. In [17] the UAV path planning has been modeled as a single objective optimization problem that utilizes a receding horizon approach and quadratic Bézier curves. The method is gradient based, allowing for quick and robust convergence to a near optimal solution. Differently, in [38] it is proposed to integrate active perception in a receding horizon setting for a goal reaching task. In particular, a perception-aware receding horizon navigation system is designed using a single forward looking camera for MAVs. In addition to avoiding obstacles, the perception-aware receding horizon navigation system is able to select motion to favor the state estimation accuracy, which is especially advantageous in environments with visually degraded regions. In [23] belief uncertainty-aware planning is followed to achieve consistent exploration. An architecture is presented to achieve this goal. It has been experimentally demonstrated that the proposed receding horizon, two-step, planning paradigm manages to explore different unknown environments with consistency, by following uncertainty optimizing trajectories that are derived through a belief-space propagation process operating on-board a Micro Aerial Vehicle robot.

*4) Bio-inspired algorithms:* In [2] an evolutionary algorithm-based path planner for UAVs has been presented, called Vibrational Genetic Algorithm (VGA), which is able to construct feasible path lines under prescribed constraints such as path length, turn angle, and clearance between the path and the boundary (terrain) within an acceptable time period, for the online planner as well, where the execution time is important. In [10] two non-deterministic algorithms are used, a Genetic Algorithm(GA) and a Particle Swarm Optimization (PSO), while in [12] a novel algorithm denoted by Harmony Search has been applied to the optimization of agricultural management tasks. In [19] a Hybrid Genetic Algorithm (HGA) is introduced to solve the path planning problem in non-convex environments. Nevertheless, HGA have given back ineffective results in terms of fuel consumption due to the high accelerations required to follow the path. In [26] a complete UAV surveillance system is designed. An Artificial Neural Network (ANN) is used to make the control structure easier to be used, to reject the disturbances and to reduce the control parameters to be controlled. Furthermore, with a combination of K-agglomerative clustering, a Set-based Particle Swarm Optimization (S-PSO) and an A* algorithm, a path can be efficiently planned predicting and reducing the energy consumption, as well. In [27], three different path planning algorithms have been implemented for biological control. The implemented algorithms are Ant Colony Optimization (ACO), Guided Local Search (GLS) and Lin-Kernighan (LKH). LKH gave better results in terms of RAM consumption, execution time, number of memory used and distance travelled.

*5) Multi-fusion based algorithms:* In [5] a hybrid meta-heuristic Ant Colony Optimization (ACO) and Differential Evolution (DE) algorithm approach is followed for Uninhabited Combat Aerial Vehicles (UCAV), allowing

for 3D path planning in combat field environments, whilst in [6] a hybrid algorithm is proposed that combines a Virtual Force (VF) algorithm for its simplicity and the A* algorithm for its robustness and efficiency. In [11] the authors have improved the PRM by random sampling in bounding boxes to ensure a more reasonable distribution in the 3D space. Based on the voxel connectivity, the selected nodes composed a roadmap, which has been applied for path searching by the A* algorithm for a feasible path. In [13] a hybrid metaheuristic approach is implemented combining GA and PSO. The disadvantage is that this approach modifies the convergence properties of the single algorithm, and the convergence of the new algorithm remains unproven. In [18] a novel approach is proposed that utilizes the formulation of dynamic Bayesian, Distance Based Value Function (DBVF) and Grey Wolf Optimization (GWO) Algorithm, while in [20] a GA has been implemented along with a Multi-Objective Path Planning (MOPP) Algorithm for area coverage and target detection. The GA aims to minimize completion time, which includes the time to find the target and the time to set up a communication path. In [21] a hybrid approach is implemented in a quadrotor. With the use of the mathematical model enriched by a Q-learning method, the goal is reached after a reinforcement learning phase. In [28] a Grey Wolf Optimization algorithm has been enriched with a Reinforcement Learning method. A cubic B-spline curve has been used to smooth the generated flight path. The authors compared the developed algorithm with another GWO algorithm, showing its statistical superiority. In [24] the chosen algorithms have been the A* and Dijkstra algorithms. Real time tests have been performed on a UAV and the two path length algorithms show the same path length. With respect to time savings, the A* algorithm showed better performance. In [40] the authors have realized a Lazy Theta* implementation for autonomous exploration in a large environment. To achieve this goal, two optimizations have been introduced taking into account the sparse grid that represents the world and the obstacle detection, while calculations are reduced by restricting the space discretization of the flight corridor to two dimensions.

### A. Challenges

Literature review reveals challenges when it comes to real time implementations that need to be overcome. Identified challenges that have motivated this research are:

1) *3D*: The proposed algorithms will be used in a 3D real world environment. Much research considers a 2.5D environment, and several existing algorithms do not account for a dynamic 3D environment.
2) *Real Time*: Since the environment in which the UAV moves is dynamically changing, the flight path may also change during flight. As such, any proposed algorithm must be real-time and on-board implementable. Many existing algorithms are not implementable online due to high computational burden.
3) *Energy Efficiency*: Power and energy limitations (fuel, battery consumption) must be considered. When it comes to aggressive maneuvers with high accelerations, this challenge becomes even more important. Managing power and energy is one of the least treated issues although as important as the other ones.
4) *Robustness*: A candidate path planning algorithm must be robust, must overcome the trap in local minima problem, must converge to at least a near optimal solution. Moreover, the algorithm must demonstrate ability to account for position sensitive device errors, rotation driving errors, linear driving errors during path planning, etc.

## III. Problem Statement

The research focuses on deriving a path planning algorithm for UAVs flying in 3D cluttered environments, which is robust, real-time implementable, it considers unknown dynamic obstacles for collision avoidance, and it is energy efficient (minimize energy consumption).

### A. 3D Path Planning

The work presented in [29], [30], [31], [32], provides a clear and mathematical definition of the 3D path planning problem. Here, fixed-wing UAVs are assumed to fly in a 3D space ($R^3$), called the workspace $w$. The workspace includes obstacles, so let $wo_i$ be the $ith$ obstacle. The free workspace without obstacles is the overall area represented by

$$w_{free} = w \setminus U_i wo_i \tag{1}$$

The initial point $x_{init}$ and the goal point $x_{goal}$ belong in the free workspace $w_{free}$. Thus, a path planning problem is defined by a triplet $(x_{init}, x_{goal}, w_{free})$.

***Definition 1***: Given is a function $\delta : [0,T] \rightarrow R^3$ of bounded variation, where $\delta(0) = x_{init}$ and $\delta(T) = x_{goal}$. If there exists a process $\Phi$ that can guarantee $\delta(\tau) \in w_{free}$ for all $\tau \in [0,T]$, then $\Phi$ is called *path planning*.

***Definition 2***: Given is a path planning problem $(x_{init}, x_{goal}, w_{free})$ and a cost function $c : \Sigma \rightarrow R \geq 0$, where $\Sigma$ denotes the set of all paths. If a process fulfills *Definition 1* to find a $\delta'$, and $c(\delta') = min\{c(\delta), \delta$ set of all feasible path$\}$, then $\delta'$ is the *optimal path* and $\Phi'$ is *optimal path planning*.

### B. Energy Computation for a Fixed-Wing Aircraft

The energy efficiency problem is accommodated as part of a node-base path planning algorithm, still being consistent with the physics of the model.
The total energy for traveling from a specified start node to a destination node is the sum of the total energy spent when travelling through the required edges. Hence, in order to calculate the energy of a path that consists of multiple straight flight paths, compute the energy consumption $\Delta E_{i,j}$ going through each straight line with distance $\Delta d$ from node $i$ to node $j$, as depicted in Figure 1. The difference in the total energy consumption is the sum of differences in potential energy and difference in kinetic

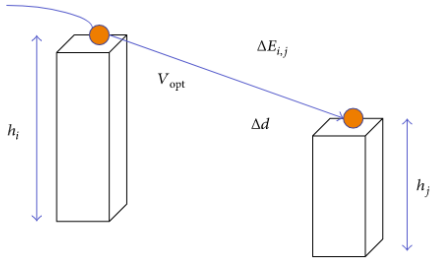


Fig. 1. Energy consumption between two nodes (picture showed in [33])

energy plus the energy used to turn the fixed-wing UAV between arcs

$$\Delta E_{i,j} = \Delta E_p + \Delta E_k + \Delta E_{turn} \tag{2}$$

Assuming there is no energy when the UAV altitude is decreasing, the difference of potential energy is given as

$$\Delta E_p = max(W\Delta h), 0) = max(mg(h_i - h_j), 0) \tag{3}$$

To calculate the kinetic energy we focus on optimizing the velocity that depends on the drag-to-lift ratio of the UAV. In [34] the drag-to-lift ratio is represented as

$$\frac{D}{L} = AV^2 + \frac{B}{V^2} \tag{4}$$

where $V$ is the flight speed and $A$ and $B$ are variables calculated by air density. The UAV parameters are given as

$$A = \frac{\rho f}{2W} \tag{5}$$

$$B = \frac{2W}{\rho b^2 \pi e} \tag{6}$$

Most of the parameters depend on the structure of the UAV; $W = mg$ is the aircraft's weight, $b$ is the wing span, $f$ is the parasite area of the aircraft, and $e$ is Oswald's efficiency. In addition, $\rho(h)$ is the air density at altitude $h$ given in [35]:

$$\rho(h) = \frac{p(h)}{RT(h)} = \frac{P_0(1 - 0.0065(h/T_0))^{5.2561}}{R(T_0 - 6.5(h/1000))} \tag{7}$$

where $P_0 = 101,325N/m^2$ is the sea level atmospheric pressure, $T_0 = 288.15K$ is the sea level standard temperature, and $R = 287.04m^2/Ks^2$ is the universal gas constant.

We assume that the fixed-wing UAV flies with a constant optimum speed $V_{opt} = (B/A)^{1/4}$, occurring when the $D/L$ ratio is optimum ($D/L_{opt} = 2\sqrt{(AB)}$). If we assume that the consumption efficiency of the UAV is 100%, then this ratio is proportional to the energy consumption, with or without the effect of any wind force.

For a zero-wind scenario, the kinetic energy $\Delta E_k$ consumed is used to overcome the drag force $D$ on a straight flight

$$\Delta E_k = \int Ddxyz = D\Delta d \tag{8}$$

Because the lift force $L$ is equal to the UAV weight, if the UAV flies with the constant optimum Carson's speed, then

$$\Delta E_k = W(D/L)_{opt}\Delta d = 2W\sqrt{\frac{f}{\bar{\rho} b^2 \pi e}}\Delta d \tag{9}$$

where

$$\bar{\rho} = (\rho(h_i) + \rho(h_j))/2 \tag{10}$$

Following the calculation from [36], the consumed energy for turning between two arcs is approximately

$$E_{turn} \approx m\frac{D}{L} sin\sigma |V^2| \tag{11}$$

where $\sigma$ is the steepest rolling angle that the drone needs to take for turning.

Parameters are computed each time instant by the function that computes the related costs, knowing the altitude $h_i$ the UAV is flying in that specific instant and knowing that:

- $g = 9.81m/s^2$, gravitational acceleration
- $P_0 = 101.325N/m^2$, sea level atmospheric pressure
- $T_0 = 288,15K$, sea level temperature
- $R = 287.04m^2/Ks^2$, universal gas constant
- $L = 0.0065K/m$, lapse rate

Thus, the following parameters can also be computed:

- $T$, the temperature at the altitude $h_i$ computed as $T_0 - Lh_i$, as long as we are in the troposphere, condition respected since the RQ-170 Sentinel is not apparently intended to flight above $15.000m$
- $P$, the atmospheric pressure at the altitude $h_i$, computed as $P = P_0(\frac{T}{T_0})^{\frac{g}{LR}}$
- $\rho(h)$, the air density at the altitude $h_i$, computed using the equation 7
- $A$ and $B$, using the equation 5, computed at the specific altitude $h_i$
- $D/L$, the drag-to-lift ratio, computed as the optimal one that is $D/L_{opt} = 2\sqrt{AB}$
- $V$, the velocity of the UAV, kept at its optimal value and computed as $v = (B/A)^{\frac{1}{4}}$
- $\sigma$, the steepest rolling angle to be taken to turn, set to a maximum of $0.78rad$ in case of turning

Given that the proposed algorithm does not have an implemented path smoothing feature but it deals with kinematic and dynamic constraints setting the maximum turning angle at $0.78rad$, the dissipated energy for a turn is not reliable, it may be misleading. Such a sharp angle for a fixed-wing UAV at high velocities implies a huge amount of energy dissipated. Turning angles in real life will be much smaller, resulting in smaller values in consumed energy for a turn. Therefore, the energy dissipated during turning maneuvers is neglected, considering only the variation of potential energy $\Delta E_p$ and the variation of the kinetic energy $\Delta E_k$.

Once the energy spent to follow the path is calculated, the power needed is obtained knowing that:

$$[J/s] = [W] \tag{12}$$

After the length of the path $d$ is computed through a proper function, the time $t$ spent to reach the goal is:

$$t = \frac{d}{V} \tag{13}$$

Knowing $t$, the power is easily obtained as:

$$P = \frac{\Delta E_{tot}}{t} \tag{14}$$

## IV. Implementation Details

The choice is to center on Node-based optimal algorithms due to their reliability and speed to find an optimal path. These characteristics make them appropriate for a 3D path planning problem.

Among them, A* is one of the best choices. Indeed, thanks to its heuristic nature, the A* search algorithm is noticeably faster than other Node-based ones. Choosing the A* algorithm two challenges are intrinsically solved: it deals with 3D environments and it is robust. The third challenge is the energy consumption problem. This is why an energy efficient version of the A* is proposed, called EEA* search algorithm. Finally, to fulfill the real-time requirement, a local path planner is added to deal with unknown dynamic obstacles. The logic is to generate intermediate way-points along the path when an obstacle is sensed by the fixed-wing UAV. The assumption that are considered are:

- Part of the map and the static obstacles of the environment are known a-priori.
- No wind or disturbances are considered during flight.

### A. A* Search Algorithm

The A* search algorithm is a node-based algorithm, an extension of Dijkstra's algorithm. As explained in [41], A* was created as part of the Shakey project that focused on building a mobile robot that could plan its own actions. Nils Nilsson originally proposed using the Graph Traverser algorithm for Shakey's path planning. Graph Traverser is guided by a heuristic function $h(x)$, the estimated distance from node $x$ to the goal node: it entirely ignores $g(x)$ the distance from the start node to $x$. Bertram Raphael suggested using the sum, $g(x) + h(x)$. Peter Hart invented the concepts we now call *admissibility* and *consistency* of heuristic functions. A* was originally designed for finding least-cost paths when the cost of a path is the sum of its edge costs, but it has been shown that A* can be used to find optimal paths for any problem satisfying the conditions of a cost algebra.

Ultimately, Node based optimal algorithms are classified because they deal with node and arc weight information; they calculate the cost by exploring the nodes to find the optimal path. Unlike Dijkstra's algorithm, A* reduces the number of states by introducing the heuristic estimation of the cost from the current state to the goal state. A* can converge very quickly and to an optimal solution.

An evaluation function is introduced, which consists of post-calculation toward the initial state and heuristic estimation toward the goal

$$f(x) = g(x) + h(x) \tag{15}$$

where g(x) is the cost from the initial state $x_{init}$ to the current state $x$, $h(x)$ is the heuristic estimation of the cost of an optimal path from the current state $x$ to the goal state $x_{goal}$.

On finite graphs with non-negative edge weights, A* is guaranteed to terminate and it is complete, i.e. it will always find a solution (a path from start to goal) if one exists. On infinite graphs with a finite branching factor and edge costs that are bounded away from zero $d(x, y) > \varepsilon > 0$ for some fixed $\varepsilon$, A* is guaranteed to terminate only if there exists a solution.

### B. EEA* Search Algorithm

Starting from the implementation of the A* search algorithm, a way to reduce as much as possible the usage of the energy is considered. This new algorithm still has all the advantages of an A* search algorithm and concurrently returns a path that is more cost-effective in terms of energy consumption.

A constant velocity along the route $V$ is assumed. The aircraft's weight $W$, its mass $m$, the wing span $b$, the parasite area of the UAV $f$, the drag-to-lift ratio $D/L$ and the Oswald's efficiency factor of the UAV $e$ are given. The distance from the node $i$ to the node $j$ $\Delta d$ is kept constant. Therefore, the parameters that affect the total energy computation are the difference of altitude $\Delta h$ in the potential energy $\Delta E_p$, the air density at a certain altitude $\rho(h)$, the kinetic energy $\Delta E_k$, and the steepest rolling angle for turning $\sigma$ that affects the energy consumed for turning, called $\Delta E_{turn}$.

In this paper, the specific parameter research has focused on because it seemed more logical to work with, is the difference of altitude $\Delta h$ between one node and its successor. In particular, limiting as much as possible the ascending movement of the fixed-wing UAV so that the total potential energy $\Delta E_p$ is as low as possible to fly from the starting node to the goal node.
Working on the kinetic energy is less relevant since the air density is strictly related to the altitude to which the UAV flies in that particular moment. Working on this parameter is harder due to some constraints in the flight. Firstly, a safety constraint on the minimum altitude to avoid collisions with the ground. Secondly, constraints inherent in laws that give certain ranges of altitudes for certain type of aircraft.

Going into details, at each computational step of the algorithm, performed in the same way as the A*, the EEA*

looks for the node with the lowest $f(x)$ in the OPEN list. Then, it confronts $z_j$, the $z$ component of the node $j$, with $z_i$, the $z$ component of the node $i$ (the parent node of $j$), obtaining two possible branches:

- if $z_j$ is equal or lower than $z_i$ then the next node has been found (the UAV is keeping is altitude or is descending).
- if $z_j$ is greater than $z_i$ then update the $jth$ node with a new greater $f(x)$ value which takes into account of the energy spent to ascend to a new altitude and repeat the process looking back in the OPEN list.

**Algorithm 1:** EEA* Search Algorithm Pseudocode

```
Result: Waypoints
put x_init in the OPEN list;
while OPEN list is not empty do
    while not done do
        take from the OPEN list the x with the
          lowest f(x) = g(x) + h(x);
        if x = x_goal then
            break;
        end
        if z component of x is lower or equal to the
          z component of its parent node then
            if auxArray is not empty then
                set the previously modified cost in
                  the OPEN list back to their
                  values;
                empty auxArray;
            end
            done;
        end
        if z component of x is greater than the z
          component of its parent node then
            if x is in auxArray then
                set the previously modified cost in
                  the OPEN list back to their
                  values;
                empty auxArray;
                done;
            else
                put x in auxArray;
                update the cost of x to the one
                  considering the ascending
                  manoeuvre;
            end
        end
    end
    same operations of the A* from now on;
end
if x different from x_goal then
    exit with error, the OPEN list is empty;
end
```

If a chosen node has been already picked up before and its $f(x)$ has already been updated, that is going to be the next node since every node can be rejected only once.
Once the next node is selected, every nodes whose $f(x)$ has been changed during this phase has brought back to its original value. This operation is needed to take into account the fact that the value of $f(x)$ changes according to the parent node where the algorithm comes from and does not depend only on the node itself. Hence, the operation has to be done again every time the algorithm has to compute a new node.

### C. Local Path Planner

When the distance of the fixed-wing goes below a certain distance with respect to an obstacle a detection happens. When this condition is verified, two different situation can be triggered:

- the UAV detects a generic dynamic obstacle: the local path planner generates an intermediate waypoint above the obstacle in order to guarantee the safety of the flight. The algorithm keep generating intermediate waypoints above the trajectory planned by the global path planner as long as the object is sensed by the sensors of the UAV.
- the UAV detects another UAV: the local path planner generates an intermediate waypoint rightwards with respect to the planned algorithm and keep doing that until the other UAV is not sensed anymore.

In the simulations a blue line is drawn to show the optimal path planned by the EEA* search algorithm, whereas a red line represents the actual trajectory performed by the fixed-wing UAV. The drone avoids in a fairly smooth manner the dynamic obstacle after sensed it, generating intermediate waypoints above the actual path computed by the global path planner. The sensing of the dynamic unknown obstacles and the computation of new waypoints is carried out in real-time.

### D. Multi-Agent Implementation

The implementation of the fleet of UAVs has been made to show that the constellation of algorithms and functions employed by each fixed-wing make them able to work properly together in a multi-agent situation. Each drone, receiving the coordinates of the starting point and of the goal point, is able to compute its own nominal path deploying the EEA* Search Algorithm; then, when the UAVs start following their route, they use their local path planning algorithm to avoid each other treating the other agents as dynamically changing obstacles. Hence an UAV does not know during the flight where the other UAVs are in a specific moment unless they are near enough to be sensed.
Furthermore, the local path planner applies different actions on the computation of the intermediate waypoint depending on if the obstacle is an unrecognized object or another agent. To do that, the agents are able to recognize the other agents once they sense each other. In 5 some examples are depicted.
The workflow of the simulation is the following one:

- The number of agents requested are spawned in the map in their respective starting node.
- The agents receive their goal node.

- The agents employ the global path planner, in the specific the EEA* Search Algorithm developed in this Thesis work.
- The simulation starts and time by time the agents monitor the area around them to look at possible threats.
- If a threat is sensed, depending on if it is an unknown obstacle or another agent, a control action is taken to make it able to avoid the collision.
- Every agent reach its goal node and the simulation ends.

## V. Simulation Results

To test the energy efficiency of the EEA* Search Algorithm 30 simulations have been carried out for the EEA* and for the A*. In order to compare the performances, the same start goals and node goals have been given as input. The tests have been made on a HP Pavillion dv6 laptop with a second generation Intel Core i7, an $8GB$ RAM and $500GB$ Hard Disk. The parameters that have been collected are the variation of potential energy $\Delta E_p$, the variation of the kinetic energy $\Delta E_k$, the variation of the total energy $\Delta E$ which is the sum of the first two, the average power consumed $P$.

Then, they have been compared to assess the performance of the EEA* in terms of energy efficiency. Specifically, the variation of the total energy has been compared and the difference between the two has been computed for every simulation done. In this way, the amount of work saved to perform the route with the novel algorithm has been obtained. After that, the same result has been computed in terms of percentage value; hence it has been gotten the percentage of variation of energy saved with the novel algorithm. Lastly, it was also interesting to do that for the average power dissipated per second, though it was not the main goal of the EEA* to minimize that.

In the table II, those computed datas have been collected and organized as follows: in the first column we have the number of the simulation which the other datas on the raw refers to, in the second column we have the total variation of energy saved, in the third column the percentage of the total variation of energy saved, in the fourth column we have the average power saved per second and in the fifth column the percentage of average power saved per second.

Looking at the table II, some conclusions can be drawn. The algorithm performs better in terms of energy consumption $86.7\%$ of the time, performs worse $10\%$ of the time and the two algorithm did the same once, equal to the $3.3\%$ of the simulation run.

When the EEA* does better, the work saved by the UAV spans between $0.13\%$ and $13.46\%$ of the one done with the A*, with an average reduction of $41.5MJ$ per flight corresponding to an average $3.43\%$ of variation of energy saved. The standard deviations are equal to $4.67MJ$ and $4.04\%$.

For what concerns the average power dissipated, the EEA* uses less power than the A* in the $86.7\%$ of the times, uses more power in the $10\%$ of the cases and uses the same power in the $3.3\%$ of the cases. Even if the percentage is the same as for the variation of the total energy, the two things does not seem to be correlated. When EEA* performs better, in percentage it consumes from $0.07\%$ to $3.73\%$ less, with an average reduction of $0.95\%$ equal to $310kW$ and with standard deviations equal to $1.11\%$ and $330kW$.

| **Sim** | $\Delta E$ **Saved** | $\Delta E\%$ **Saved** | $P$ **Saved** | $P\%$ **Saved** |
|---|---|---|---|---|
| 1 | $0.007GJ$ | $0.53\%$ | $0.97MW$ | $3.22\%$ |
| 2 | $-0.059GJ$ | $-4.63\%$ | $0.59MW$ | $1.97\%$ |
| 3 | $0.020GJ$ | $1.85\%$ | $1.18MW$ | $3.73\%$ |
| 4 | $0.004GJ$ | $0.30\%$ | $-0.31MW$ | $-1.05\%$ |
| 5 | $0.007GJ$ | $0.52\%$ | $0.02MW$ | $0.07\%$ |
| 6 | $0.011GJ$ | $1.20\%$ | $0.89MW$ | $2.65\%$ |
| 7 | $0.0081GJ$ | $1.15\%$ | $0.53MW$ | $1.47\%$ |
| 8 | $0.0151GJ$ | $2.79\%$ | $-0.02MW$ | $-0.05\%$ |
| 9 | $0.0961GJ$ | $6.07\%$ | $0.19MW$ | $0.65\%$ |
| 10 | $0.0918GJ$ | $6.36\%$ | $0.54MW$ | $1.81\%$ |
| 11 | $0.0905GJ$ | $5.15\%$ | $0.09MW$ | $0.31\%$ |
| 12 | $0.0082GJ$ | $0.63\%$ | $0.51MW$ | $1.68\%$ |
| 13 | $0.0019GJ$ | $0.13\%$ | $0.18MW$ | $0.61\%$ |
| 14 | $-0.0195GJ$ | $-1.40\%$ | $0.18MW$ | $0.61\%$ |
| 15 | $0.0923GJ$ | $5.58\%$ | $-0.01MW$ | $-0.03\%$ |
| 16 | $0.0765GJ$ | $6.56\%$ | $0.36MW$ | $1.14\%$ |
| 17 | $0.0863GJ$ | $11.61\%$ | $0.24MW$ | $0.64\%$ |
| 18 | $0GJ$ | $0\%$ | $0MW$ | $0\%$ |
| 19 | $-0.0229GJ$ | $-1.61\%$ | $0.78MW$ | $2.64\%$ |
| 20 | $0.0554GJ$ | $4.61\%$ | $0.28MW$ | $0.92\%$ |
| 21 | $0.0139GJ$ | $1.03\%$ | $0.08MW$ | $0.27\%$ |
| 22 | $0.0319GJ$ | $2.20\%$ | $0.06MW$ | $0.21\%$ |
| 23 | $0.0912GJ$ | $6.64\%$ | $0.05MW$ | $0.17\%$ |
| 24 | $0.0404GJ$ | $2.30\%$ | $0.05MW$ | $0.18\%$ |
| 25 | $0.0661GJ$ | $4.69\%$ | $0.40MW$ | $1.36\%$ |
| 26 | $0.0856GJ$ | $6.69\%$ | $0.23MW$ | $0.77\%$ |
| 27 | $0.0585GJ$ | $4.24\%$ | $0.20MW$ | $0.67\%$ |
| 28 | $0.0389GJ$ | $3.36\%$ | $0.24MW$ | $0.77\%$ |
| 29 | $0.1225GJ$ | $13.46\%$ | $0.39MW$ | $1.13\%$ |
| 30 | $0.1276GJ$ | $10.90\%$ | $0.46MW$ | $1.60\%$ |

TABLE II

Results of the comparison between A* and EEA*. **Sim** stands for the number of simulation, $\Delta E$ **Saved** is the work saved employing the EEA*, $\Delta E\%$ **Saved** is the percentage of work saved, $P$ **Saved** is the amount of average power saved, $P\%$ **Saved** is the percentage of average power saved.

| |
|---|
| $\Delta E$ **Saved** |
| Mean $= 41.5MJ$ |
| Standard Deviation$= 46.8MJ$ |
| Mean$[\%] = 3.43\%$ |
| Standard Devation$[\%] = 4.04\%$ |
| $P$ **Saved** |
| Mean$= 310kW$ |
| Standard Deviation$= 330kW$ |
| Mean$[\%] = 0.95\%$ |
| Standard Deviation$[\%] = 1.11\%$ |

TABLE III

The table above collects the mean value of the variation of the total energy which is saved, its standard deviation and those values expressed in percentage. Below, the mean value of the average power saved, its standard deviation and those values expressed in percentage.

During the 30 simulations using the A* and the 30 simulations of the EEA*, for a total of 60 simulations, no collision has been registered. Both the static obstacles and the dynamic obstacles has been avoided correctly by the algorithms reporting good performances.

In the figure 4 is depicted an example of the control action applied by the local path planner. The diagram depicted shows on the vertical axis the distance expressed in meters between an UAV and one of the dynamic threats present in the map.

As it can be seen, the UAV approaches the object quickly

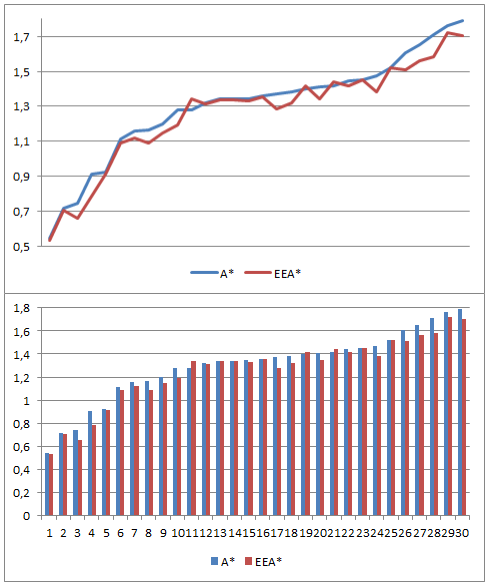


Fig. 2. Comparison between A* and EEA* in terms of variation of total energy consumption. The first picture shows the results plotted on a diagram. The second picture shows the results plotted on a histogram. Along the y-axis there is the percentage value, along the x-axis the number of the simulation associated to that value for both the plots.

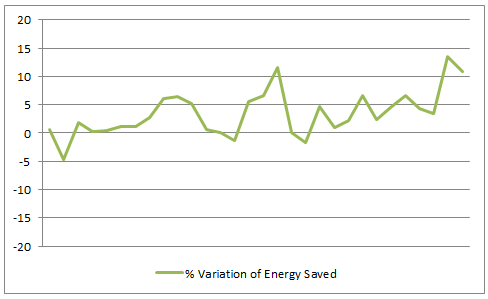


Fig. 3. The plot shows the percentage of variation of the total energy saved by means of using the EEA* algorithm instead of the A* along the y-axis and the number of the simulation on the x-axis.

and dangerously in the first part of the plot. Nonetheless, at a certain point, when the distance is more or less near to $500m$, the sensors of the drone detect the moving obstacle approaching and invading the area right in front of the fixed-wing UAV. At that point, the local path planner comes into action. Moving the UAV upwards, the local path planner allows the aircraft to be kept far from the threats avoiding a collision. This behaviour is depicted in the diagram with the sharp turning of the function, with the distance kept for a while at a reasonable distance of half a kilometer. In that area of the plot the UAV and the object are flying in the same region, but thanks to the local path planner, they do that in a safe way. After that, it can be seen an almost linear increase of the distance between the two, signifying a departure since both the UAV and the object keep flying on their own path.

## VI. Conclusions and Future Works

In this research work has been done a literature review concerning the last decade researches on path planning for UAVs. After a classification of the algorithms employed, limitations have been identified and common challenges

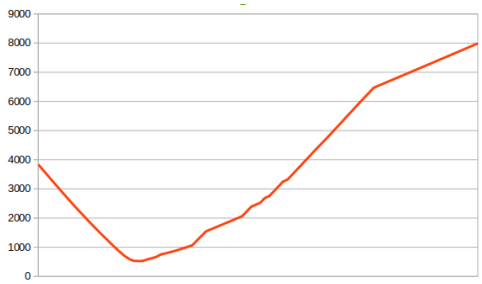


Fig. 4. The diagram shows the action of the local path planner with respect to the distance from the UAV and the object. The y-axis represents the distance in meters between a certain UAV and a dynamic threats in the map, the x-axis the time.

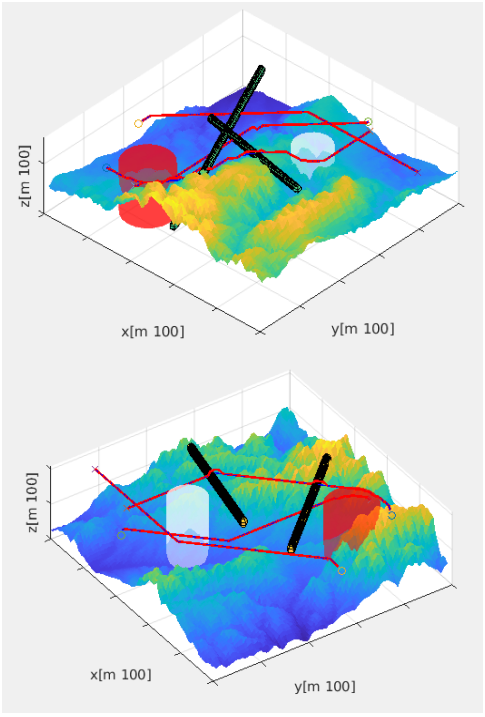


Fig. 5. Fleet of UAVs performing their path altogether. When the path intersect the object is not actual a collision but it means the object or the UAV are passed in a different moment in time so they did not collide at all.

have been recognized. These challenges have been overcome in different ways by researchers, nonetheless, as far as we know, none of them have solved the four problems altogether, at leas for UAV applications.

Later, a novel node-based 3D real-time energy efficient path planning algorithm has been developed using MATLAB in order to overcome this four challenges altogether. Specifically, the algorithm, called EEA* Search Algorithm, is based on the A* Search Algorithm and deploys a global path planner and a local path planner finding the most energy efficient route and avoiding dynamic unknown obstacles in the map.

The algorithms have been deployed in different and gradually more complicated environments. Then, three fixed-wings UAVs have been employed to work together without interfere with each other.

The energy efficiency has been tested. To do that, 30 simulations have been carried out for both the EEA* and

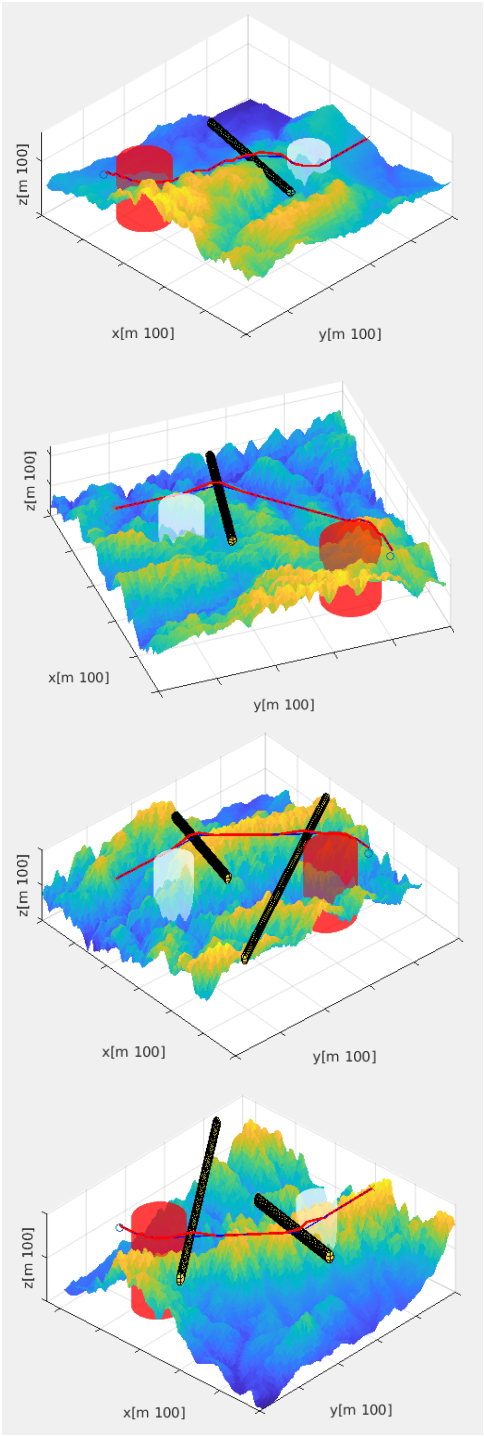


Fig. 6. Simulations in different mountain environments. In every map there is a no-fly area in red and a dangerous weather area in white. The blue line is the one computed by the global path planner, the red line is the actual trajectory performed by the fixed-wing UAV. The black and yellow objects are the dynamic threats. The objects are spheres whose the head is the actual threat; the tail is not a threat, but it is represented to show the trajectory followed by the dynamically changing obstacles. The small circle is the starting node, the small cross is the goal node.

the A* feeding them with the same inputs and looking for the results. From the datas collected it comes out that the EEA* reduces the work needed to follow the route in 86.7% of the time and of the 3.43% on average. Furthermore, also the average power dissipated has been indirectly decreased in 86.7% of the cases of a 0.95%.

Even if the algorithm seems to show a good behaviour, still a lot of future work has to be done. In future works, we have to aim at increasing the uncertainties adding wind disturbance in such a way to work on an environment as much as similar to the real world as possible. Another important aspect to be considered, to better validate the algorithm developed, is to add a path smoothing algorithm in order to have more realistic and feasible paths to be performed by a fixed-wing UAV. This feature not only have this pro, but it would permit us to study better the performance of the energy efficiency, taking into account of the work needed for turning maneuvers. Indeed, with more realistic paths, the variation of the energy would be much more accurate, further reducing the approximation of the results.

## ACKNOWLEDGMENT

A.R. acknowledges Compagnia di San Paolo for financial support.

## REFERENCES

[1] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, Y. Xia, *Survey of Robot 3D Path Planning Algorithms*, Hindawi Publishing Corporation Journal of Control Science and Engineering Volume 2016, Article ID 7426913, 22 pages, 2016. http://dx.doi.org/10.1155/2016/7426913, 2016.

[2] Y.V. Pehlivanoglu, O. Baysal, A. Hacioglu, *Path planning for autonomous UAV via vibrational genetic algorithm*, Aircraft Engineering and Aerospace Technology, Vol. 79 Iss 4 pp. 352 - 359, 2007.

[3] K. Yang, S. Sukkarieh, *3D Smooth Path Planning for a UAV in Cluttered Natural Environments*, 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.

[4] D. Zhuoninga, Z. Rulinb, C. Zongjia, Z. Ruia, *Study on UAV Path Planning Approach Based on Fuzzy Virtual Force*, Chinese Journal of Aeronautics 23(2010) pp. 341-350, 2010.

[5] H. Duan, Y. Yu, X. Zhang, S. Shao, *Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm*, Simulation Modelling Practice and Theory 18 (2010) 1104–1115, 2010.

[6] Z.Dong, Z. Chen, R. Zhou, R. Zhang, *A Hybrid Approach of Virtual Force and A* Search Algorithm for UAV Path Re-Planning*, 2011 6th IEEE Conference on Industrial Electronics and Applications, 2011.

[7] E.I. Grøtli, T.A. Johansen, *Path Planning for UAVs Under Communication Constraints Using SPLAT! and MILP*, J Intell Robot Syst (2012) pp. 265–282, 2012.

[8] A. Chamseddine, Y. Zhang, C.A. Rabbath, C.Join, D. Theilliol, *Flatness-Based Trajectory Planning/Replanning for a Quadrotor Unmanned Aerial Vehicle*, IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS VOL. 48, NO. 4 OCTOBER 2012, 2012.

[9] L. Wei, Z. Zheng, C. Kai-Yuan, *Bi-level programming based real-time path planning for unmanned aerial vehicles*, Knowledge-Based Systems 44 (2013) pp. 34–47, 2013.

[10] V. Roberge, M. Tarbouchi, G. Labonté, *Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning*, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 9, NO. 1, FEBRUARY 2013, 2013.

[11] F. Yan, Y. Liu, J.Z. Xiao, *Path Planning in Complex 3D Environments Using a Probabilistic Roadmap Method*, International Journal of Automation and Computing 10(6), December 2013, pp. 525-533, 2013.

[12] J. Valente, J. Del Cerro, A. Barrientos, David Sanz, *Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach*, Computers and Electronics in Agriculture 99 (2013) pp. 153–159, 2013.

[13] V. Roberge, M. Tarbouchi, F. Allaire, *PARALLEL HYBRID META-HEURISTIC ON SHARED MEMORY SYSTEM FOR REAL-TIME UAV PATH PLANNING*, International Journal of Computational Intelligence and Applications Vol. 13, No. 2 (2014) 1450008 (16 pages), 2014.

[14] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, *Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots*, Auton Robot, 2015.

[15] D. Ortiz-Arroyo, *A Hybrid 3D Path Planning Method for UAVs*, 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS) November 23-25, 2015. Cancun, Mexico, 2015.

[16] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, *Receding horizon path planning for 3D exploration and surface inspection* Auton Robot (2018) 42 pp. 291–306, 2016.

[17] B. Ingersoll, K. Ingersoll, P. DeFranco, A. Ningx, *UAV Path-Planning using Bézier Curves and a Receding Horizon Approach*, AIAA Modeling and Simulation Technologies Conference, 2016.

[18] M. Radmanesh, M. Kumar, *Grey Wolf Optimization based Sense and Avoid Algorithm for UAV Path Planning in Uncertain Environment using a Bayesian Framework*, 2016 International Conference on Unmanned Aircraft Systems (ICUAS) June 7-10, 2016. Arlington, VA USA, 2016.

[19] M. da Silva Arantes, J. da Silva Arantes C. F. M. Toledo, B. C. Williams, *A Hybrid Multi-Population Genetic Algorithm for UAV Path Planning*, GECCO '16, July 20-24, 2016, Denver, CO, USA, 2016.

[20] S. Hayat, E. Yanmaz, T. X. Brown, C. Bettstetter, *Multi-Objective UAV Path Planning for Search and Rescue*, 2017 IEEE International Conference on Robotics and Automation (ICRA) Singapore, May 29 - June 3, 2017, 2017.

[21] T. Zhang, X. Huo, S. Chen, B. Yang, G. Zhang, *Hybrid Path Planning of A Quadrotor UAV Based on Q-Learning Algorithm*, Proceedings of the 37th Chinese Control Conference July 25-27, 2018, Wuhan, China, 2018.

[22] G. Zhang, L. Hsu, *A New Path Planning Algorithm Using a GNSS Localization Error Map for UAVs in an Urban Area*, Journal of Intelligent & Robotic Systems, 2019.

[23] C. Papachristos, F. Mascarich, S. Khattak, T. Dang, K. Alexis, *Localization uncertainty-aware autonomous exploration andmapping with aerial robots using receding horizon path-planning.*, Autonomous Robots https://doi.org/10.1007/s10514-019-09864-1, 2019.

[24] E. J. Dhulkefl, A. Durdu, *Path Planning Algorithms for Unmanned Aerial Vehicles*, International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-

[25] X. Liu, X. Du, X. Zhang, Q. Zhu, M. Guizani, *Evolution-algorithm-based unmanned aerial vehicles path planning in complex environment*, Computers and Electrical Engineering 80 (2019) 106493, 2019.

[26] R. J. Wai, A. S. Prasetia, *Adaptive Neural Network Control and Optimal Path Planning of UAV Surveillance System With Energy Consumption Prediction*, IEEE Access VOLUME 7, 2019, 2019.

[27] H. Freitas, B. S. Faiçal, A. Vinicius Cardoso e Silva, J. Ueyama, *Use of UAVs for an efficient capsule distribution and smart path planning for biological pest control*, Computers and Electronics in Agriculture 173 (2020) 105387, 2020.

[28] C. Qu, W. Gai, M. Zhong, J. Zhang, *A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning*, Applied Soft Computing Journal (2020), doi: https://doi.org/10.1016/j.asoc.2020.106099, 2020.

[29] Choset and M. Howie, *Principles of robot motion: theory algorithms and implementations*, MIT press, 2005.

[30] Schøler and Flemming, *3D Path Planning for Autonomous Aerial Vehicles in Constrained Spaces*, Diss. Videnbasen for Aalborg UniversitetVBN Aalborg UniversitetAalborg University Det Teknisk-Naturvidenskabelige FakultetThe Faculty of Engineering and Science Institut for Elektroniske SystemerDepartment of Electronic Systems.

[31] S M LAValle, *Planning algorithms*, Cambridge university press, 2006.

[32] S Karaman and E. Frazzoli, *Sampling-based algorithms for optimal motion planning*, The International Journal of Robotics Research, vol. 30, no. 7, pp. 846-894, 2011.

[33] Z. Ahmad, F. Ullah, C. Tran, S. Lee, *Efficient Energy Flight Path Planning Algorithm Using 3-D Visibility Roadmap for Small Unmanned Aerial Vehicle*, International Journal of Aerospace Engineering, 2017.

[34] B. H. Carson, *Fuel efficiency of small aircraft*, Journal of Aircraft, vol. 19, no. 6, pp. 473–479, 1982.

[35] M. Cavcar, *The international standard atmosphere (ISA)*, Anadolu University, Turkey, vol. 30, 2000.

[36] G. Nachmani, *Minimum-energy flight paths for UAVs using mesoscale wind forecasts and approximate dynamic programming*, DTIC Document, 2007.

[37] L. Wei, Z. Zheng, C. Kaiyuan, *Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval*, Chinese Journal of Aeronautics, 2013,26(3) pp. 646–660, 2013.

[38] Z. Zhang, D. Scaramuzza, *Perception-aware Receding Horizon Navigation for MAVs*, 2018 IEEE International Conference on Robotics and Automation (ICRA) May 21-25, 2018, Brisbane, Australia, 2018.

[39] M. Kang, Y. Liu, Y. Ren†, Y. Zhao, Z. Zheng, *An Empirical Study on Robustness of UAV Path Planning Algorithms Considering Position Uncertainty*, 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), 2017.

[40] M. Faria, R. Marín, M. Popovic, I. Maza, A. Viguria, *Efficient Lazy Theta* Path Planning over a Sparse Grid to Explore Large 3D Volumes with a Multirotor UAV*, Sensors 2019, 19, 174, 2019.

[41] https://en.wikipedia.org/wiki/A*_search_algorithm