



Fixed-Wing UAV Energy Efficient 3D Path Planning in Cluttered Environments

Giuseppe Aiello¹ · Kimon P. Valavanis² · Alessandro Rizzo³

Received: 15 April 2021 / Accepted: 1 March 2022 / Published online: 1 July 2022
© The Author(s) 2022, corrected publication 2022

Abstract

UAV path planning in 3D cluttered and uncertain environments centers on finding an optimal / sub-optimal collision-free path, considering in parallel geometric, physical and temporal constraints, for example, obstacles, infrastructure, physical or artificial landmarks, etc. This paper introduces a novel node-based algorithm, called Energy Efficient A* (EEA*), which is based on the A* search algorithm, but overcomes some of its key limitations. The EEA* deals with 3D environments, it is robust converging fast to the solution, it is energy efficient and it is real-time implementable and executable. In addition to the EEA*, a local path planner is also derived to cope with unknown dynamic threats within the working environment. The EEA* and the local path planner are first implemented and evaluated via simulated experiments using a fixed-wing UAV operating in mountain-like 3D environments, and in the presence of unknown dynamic obstacles. This is followed by evaluating a set up where three UAVs are commanded to follow their respective paths in a safe way. The energy efficiency of EEA* is also tested and compared with the conventional A* algorithm.

Keywords Fixed-wing UAV · 3D path planning · Node-based algorithm · Energy efficiency

1 Introduction

Without loss of generality, simple 2D path planning algorithms are not implementable in complex 3D uncertain environments. On the other hand, challenges in 3D path planning are basically due to their computational complexity, underlying vehicle physical, kinematic and dynamic constraints, vehicle maneuverability, to say the least. Derivation and implementation of a collision free 3D path planning algorithm in cluttered environments constitutes an NP-hard

problem. There is no universally acceptable solution, and most of existing solutions are not optimal but heuristic ones.

This paper proposes a novel EEA* algorithm and a local path planner that are real-time implementable in 3D cluttered environments; they account for dynamic collision avoidance and they consider physical constraints and limitations of the fixed-wing UAV. The EEA* computes the energy consumed by the vehicle (in this case, the UAV) while following its path, and it minimizes this value by adapting the path, accordingly. The local path planner copes in real-time with unknown dynamic threats present in the operational space, after an off-line computation of the nominal path that is to be followed. It is shown that the energy efficiency of the proposed EEA* algorithm is superior to the conventional A* algorithm; the EEA* reduces on average the energy needed to follow the route by 3.43% in 86.7% of the tested case studies.

The rest of the paper is organized as follows. Section II presents a well-focused literature review, with emphasis on listing advantages, disadvantages, implementation challenges, and limits of existing path planning techniques. Section III states the problem and derives the proposed EEA* algorithm, while Section IV discusses implementation details. Section V presents a comprehensive simulation

✉ Alessandro Rizzo
alessandro.rizzo@polito.it

Giuseppe Aiello
s263610@studenti.polito.it

Kimon P. Valavanis
kimon.valavanis@du.edu

¹ Politecnico di Torino, 10129 Torino TO, Italy

² Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO 80210 USA

³ Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino TO, Italy

campaign that supports the effectiveness of the approach. Section VI concludes the paper.

2 Literature Review

The UAV path planning problem is cast as an optimization problem that returns an optimal solution subject to constraints. Regardless, the specific challenges that have motivated this research, and have dictated the literature review are:

1. *3D*: Any proposed algorithm should 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 vehicle (UAV) moves is/may be 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*: Energy limitations (fuel, battery consumption) must be considered. When it comes to aggressive maneuvers with high accelerations, this challenge becomes even more important. Managing 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.

When looking at published literature, according to [1], path planning algorithms may be divided in the following five categories that are reviewed next.

2.1 Sampling Based Algorithms

In [3] and [14], two versions of the same algorithm have been used: RRT and RRT*. In the former, 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 latter, 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, 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 fashion through a fuzzy controller working simultaneously with a Lazy Theta* algorithm [40]. The best path may be found to avoid obstacles by changing the UAV's altitude online using data from the sensors and topographical database.

2.2 Node Based Optimal Algorithms

Node based optimal algorithms model and explore routes between nodes of a graph, representing the navigable space, starting at one node and traversing nodes until the destination has been reached. These algorithms have been successfully used to identify optimal routes through a graph for uses such as logistics planning, least cost call or IP routing, and gaming simulation.

In [4], a Fuzzy Virtual Forces (FVF) algorithm is used. To solve the issue of local minima encountered in VF algorithms, the combination of threats has been integrated in the algorithm through a suitable adjacency matrix. 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 implementing the GNSS, whereas in [25], the authors have proposed an Evolutionary Optimization Algorithm based on an improved t-distribution to deal with high computational complexity.

2.3 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. However, using such a simplified models in trajectory planning increased uncertainties and mismatch with the real system. In [9], a Bi-Level Programming 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 through a receding horizon technique. In [17], the UAV path planning problem 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 verified 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.

2.4 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 provided 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.

2.5 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 [39] 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 layout, while calculations are reduced by restricting the space discretization of the flight corridor to two dimensions.

2.6 Summary

Table 1 summarizes key characteristics of each reviewed work. It is important to emphasize that limitations refer to the implementation presented in the reviewed paper. The reviewed algorithms may not have been used in the cited references to overcome these limitations; however,

Table 1 Summary of UAV path planning review

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

Columns contain referenced work and year of publication (REF), UAV Type, developed/implemented algorithms (ALGORITHM), challenges not satisfied, numbered as previously described (LIMITS)

this does not necessarily mean that they are incapable of doing that. For example, MILP algorithms may be used to solve 3D path planning problems. But a MILP algorithm implemented in a 3D environment requires a heavy computational burden. Developing an ad-hoc solution to overcome these challenges may result in a better solution.

3 Problem Statement

The problem statement is detailed in this Section.

3.1 3D Path Planning

An accurate mathematical definition of the 3D path planning problem is presented in [29–32]. Here, fixed-wing UAVs are

assumed to fly in a 3D space (\mathbb{R}^3), called the workspace w . The workspace includes obstacles, so let w_{o_i} be the i th obstacle. The free workspace without obstacles is the region represented by

$$w_{\text{free}} = w \setminus \bigcup_i w_{o_i} \quad (1)$$

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

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

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

3.2 Energy Computation for a Fixed-Wing Aircraft

The energy efficiency problem is accommodated as part of a node-based 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 traveling through the required edges. Hence, in order to calculate the energy of a path that consists of multiple straight flight paths, we compute the energy consumption $\Delta E_{i,j}$ flying through each straight line of length Δd from node i to node j . As explained in details in [33], the difference in the total energy consumption is the sum of differences in potential energy and difference in kinetic energy, plus the energy used to turn the fixed-wing UAV

$$\Delta E_{i,j} = \Delta E_p + \Delta E_k + \Delta E_{\text{turn}} \quad (2)$$

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

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

where h_i and h_j are the altitudes of node i and j , respectively.

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 computed as

$$\frac{D}{L} = AV^2 + \frac{B}{V^2}, \quad (4)$$

where V is the flight speed and A and B are UAV parameters, also depending on the air density. The UAV parameters are given as

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

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

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 , computed as [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 Carson’s speed $V_{opt} = (B/A)^{1/4}$, occurring when the D/L ratio is optimum ($D/L_{opt} = 2\sqrt{AB}$). Assuming that the consumption efficiency of the UAV is 100%, then this ratio is proportional to the energy consumption, regardless the effect of any wind force.

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

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

Since 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 in [36], the consumed energy for turning between two arcs is approximately

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

where σ is the steepest rolling angle that the drone needs to take for turning.

Parameters are computed at each time instant within the computation of the cost, assuming that the altitude h_i at which the UAV is flying in that specific instant is known, and assuming:

- $g = 9.81m/s^2$, 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

Hence, 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 Eq. 7;
- A and B , using Eq. 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 $v = (B/A)^{\frac{1}{4}}$;
- σ , the steepest rolling angle to be taken to turn, set to a maximum of 0.78 rad 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 0.78 rad, the dissipated energy for a turn may not be reliable, sometimes 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, while we explicitly consider the variation of potential energy ΔE_p and the variation of the kinetic energy ΔE_k .

4 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.

After an offline phase, which can be performed while the aircraft is on ground in order to not largely harm the performance, where the global path planner evaluate the a-priori known part of the map, the online part is suitable for online implementation. During the online stage the local path planner evaluates dynamically the unknown part of the maps, the dynamically changing obstacles and the other fixed-wing UAVs flying in the map, computing the path accordingly.

Algorithm 1 A* Search Algorithm Pseudocode.

```

Result: Waypoints
put  $x_{init}$  in the OPEN list;
while OPEN list is not empty do
  take from the OPEN list the  $x$  with the lowest
   $f(x) = g(x) + h(x)$ ;
  if  $x = x_{goal}$  then
    | break;
  end
  generate each node successors  $x_{succ}$  that come after  $x$ 
  for each  $x_{succ}$  of  $x$  do
    successor current cost =  $g(x) + w(x, x_{succ})$ ;
    if  $x_{succ}$  is in the OPEN list then
      | if successor current cost  $\leq g(x_{succ})$  then
        | | jump outside the for;
      | end
    else if  $x_{succ}$  is in the CLOSED list then
      | if successor current cost  $\leq g(x_{succ})$  then
        | | move  $x_{succ}$  from the CLOSED list to
        | | the OPEN list;
      | end
    else
      | add  $x_{succ}$  to the OPEN list;
      | set the parent of  $x_{succ}$  to  $x$ ;
    end
     $g(x_{succ}) =$  successor current cost;
    set the parent of  $x_{succ}$  to  $x$ ;
  end
  add  $x$  to the CLOSED list;
end
if  $x$  different from  $x_{goal}$  then
  | exit with error, the OPEN list is empty;
end

```

4.1 A* Search Algorithm

The A* search algorithm is a node-based algorithm, an extension of Dijkstra's algorithm. As explained in [https://en.](https://en.wikipedia.org/wiki/A*_search_algorithm)

[wikipedia.org/wiki/A*_search_algorithm](https://en.wikipedia.org/wiki/A*_search_algorithm), 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

$$f(x) = g(x) + h(x) \quad (12)$$

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 ε , A* is guaranteed to terminate only if there exists a solution.

4.2 The Proposed 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 Δd is kept constant. Therefore, the parameters that affect the total energy computation are the difference of altitude Δh in the

potential energy ΔE_p , the air density at a certain altitude $\rho(h)$, the kinetic energy ΔE_k , and the steepest rolling angle for turning σ that affects the energy consumed for turning, called ΔE_{turn} .

Here, the specific parameter research has focused on because it seemed more logical to work with, is the difference of altitude Δh 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 ΔE_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 its altitude or is descending).
- if z_j is greater than z_i then update the j th 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.

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.

4.3 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.

Algorithm 2 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

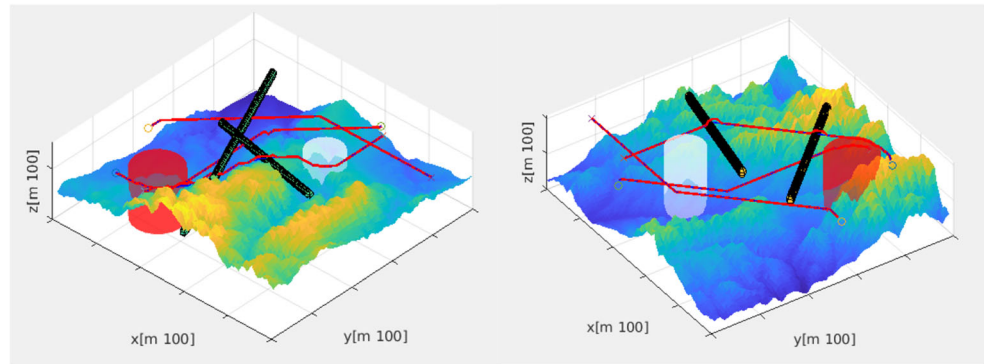
```

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

Fig. 1 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



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.

4.4 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.

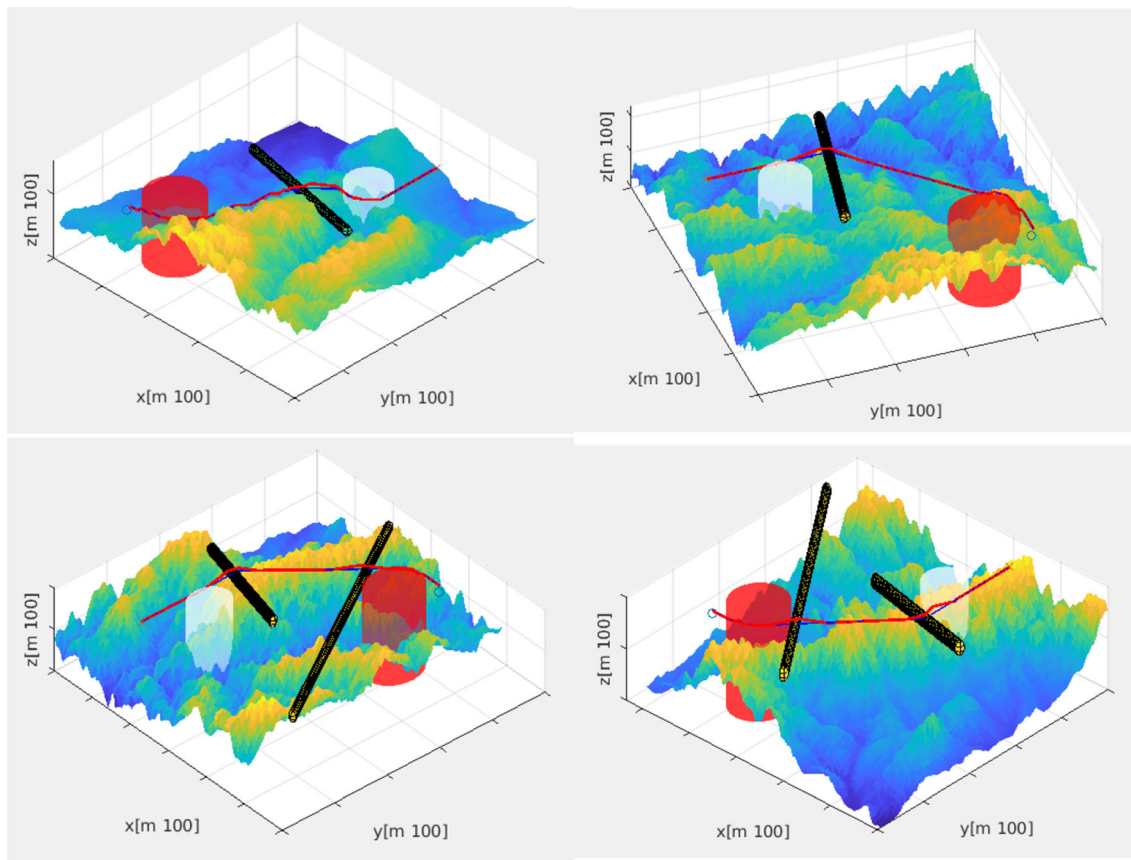


Fig. 2 Simulations in different mountain environments. In every map there is a no-fly area in red and a dangerous weather area in white. The blue line is the one computed by the global path planner, the red line is the actual trajectory performed by the fixed-wing UAV. The black and

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

Table 2 Results of the comparison between A* and EEA*. **Sim** stands for the number of simulation, ΔE **Saved** is the work saved employing the EEA*, $\Delta E\%$ **Saved** is the percentage of work saved

Sim	ΔE Saved	$\Delta E\%$ Saved
1	0.007GJ	0.53%
2	-0.059GJ	-4.63%
3	0.020GJ	1.85%
4	0.004GJ	0.30%
5	0.007GJ	0.52%
6	0.011GJ	1.20%
7	0.0081GJ	1.15%
8	0.0151GJ	2.79%
9	0.0961GJ	6.07%
10	0.0918GJ	6.36%
11	0.0905GJ	5.15%
12	0.0082GJ	0.63%
13	0.0019GJ	0.13%
14	-0.0195GJ	-1.40%
15	0.0923GJ	5.58%
16	0.0765GJ	6.56%
17	0.0863GJ	11.61%
18	0GJ	0%
19	-0.0229GJ	-1.61%
20	0.0554GJ	4.61%
21	0.0139GJ	1.03%
22	0.0319GJ	2.20%
23	0.0912GJ	6.64%
24	0.0404GJ	2.30%
25	0.0661GJ	4.69%
26	0.0856GJ	6.69%
27	0.0585GJ	4.24%
28	0.0389GJ	3.36%
29	0.1225GJ	13.46%
30	0.1276GJ	10.90%

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 Fig. 1 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.

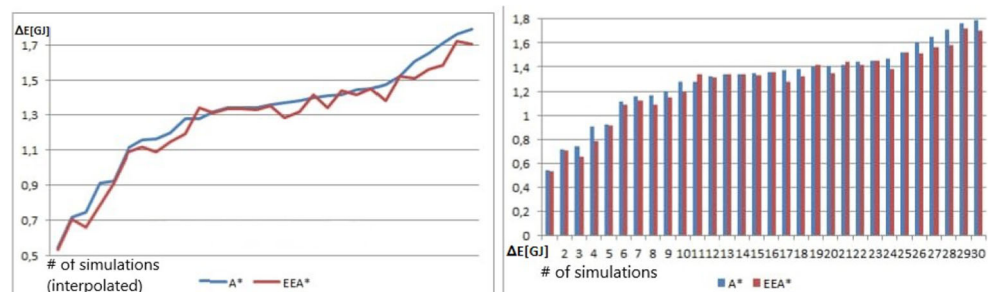
5 Simulation Results

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 ΔE_p , the variation of the kinetic energy ΔE_k , the variation of the total energy ΔE which is the sum of the first two. 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. In Figs. 1 and 2 the simulations are showed. Simulations have been performed

Fig. 3 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



in different mountain-wise environments and both in normal and multi-agent implementations.

In the Table 2, 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.

Looking at the Table 2, 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. In Figs. 3 and 4 the results are depicted via diagrams and histograms.

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%.

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 Table 3, the performances are showed in terms of mean value of the variation of the total energy saved, its standard deviation and those values expressed in percentage.

In the Fig. 5 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

Table 3 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

ΔE Saved
Mean = 41.5MJ
Standard Deviation= 46.8MJ
Mean[%] = 3.43%
Standard Deviation[%] = 4.04%

between an UAV and one of the dynamic threats present in the map.

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

Fig. 4 The plot shows the percentage of variation of the total energy saved by means of using the EEA* algorithm instead of the A*

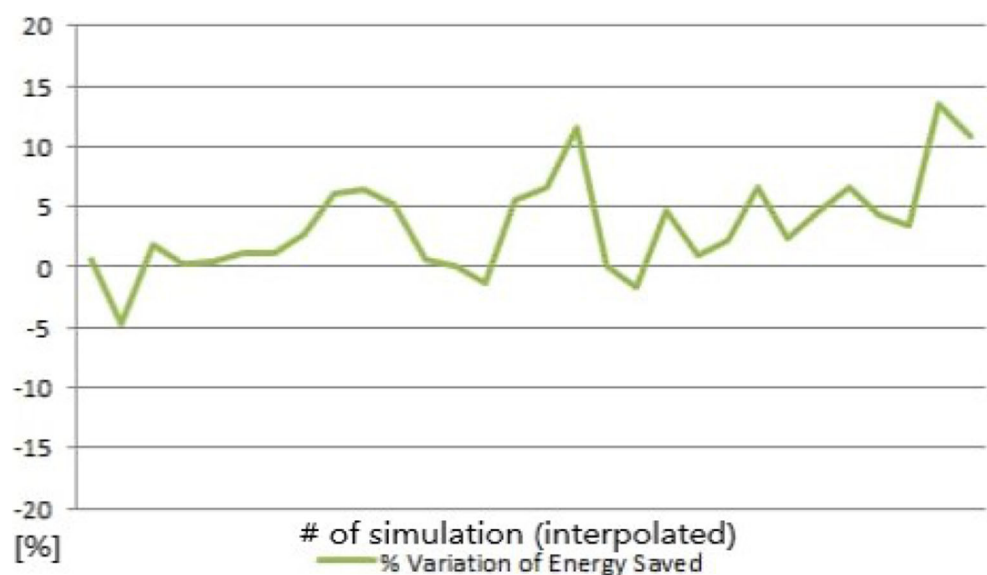
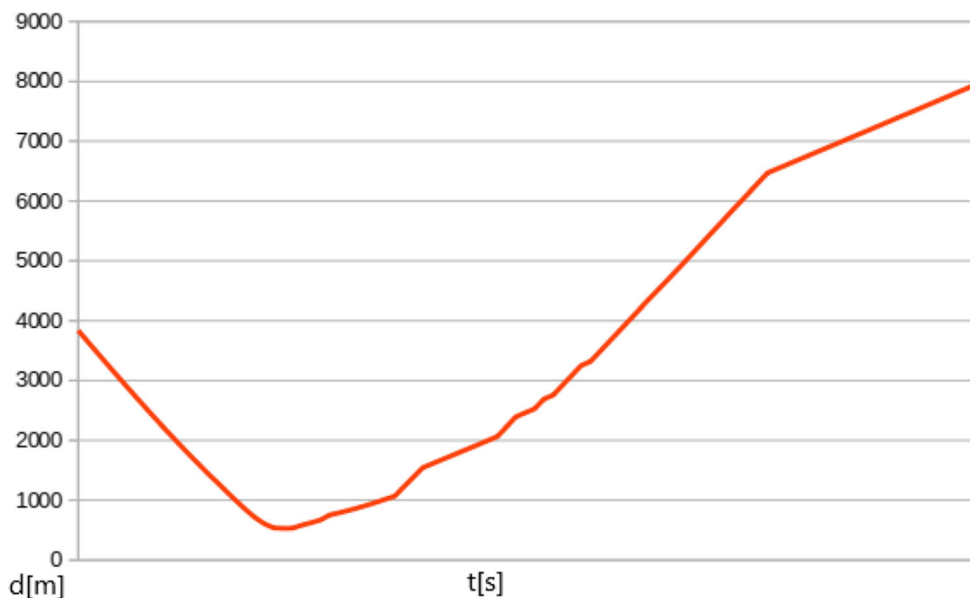


Fig. 5 The diagram shows the action of the local path planner with respect to the distance from the UAV and the object. The vertical axis represents the distance in meters between a certain UAV and a dynamic threats in the map



6 Conclusions and Future Work

In this work, we have firstly carried out 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 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 least 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 these 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 the A* feeding them with the same inputs and looking for the results. From the data collected, we observe that the EEA* reduces the work needed to follow the route in 86.7% of the time and of the 3.43% on average.

Even if the algorithm provides encouraging preliminary results, some improvements are required. In future works, we will aim at increasing the uncertainty sources, contemplating wind disturbance toward working in an environment similar to the real world. Another important aspect to be

considered, to better validate the algorithm developed, is to include 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 advantage, but it would permit us to better study the performance of the energy efficiency, taking into account of the work needed for turning maneuvers. Indeed, with more realistic paths, the estimation of the variation of energy would be much more accurate, further reducing the approximation of the results.

Acknowledgements This paper is an extended and enhanced version of the conference paper "3D Real-Time Energy Efficient Path Planning for a Fleet of Fixed-Wing UAVs", published in Proceedings, 2021 International Conference on Unmanned Aircraft Systems, June 2021, Athens, Greece.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement. A.R. acknowledges partial funding by Compagnia di San Paolo and the 2019 Amazon Research Award.

Author Contributions K.V. and A.R. conceived and supervised the research. G.A. designed and implemented the simulation studies and drafted a preliminary version of the paper. All the authors concurred to the analysis of data, revised, and approved the final version of the manuscript.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not

included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Yang, L., Qi, J., Song, D., Xiao, J., Han, J., Xia, Y.: Survey of Robot 3D Path Planning Algorithms. Hindawi Publishing Corporation Journal of Control Science and Engineering **2016**(7426913, 22), 2016 (2016). <https://doi.org/10.1155/2016/7426913>
2. Pehlivanoglu, Y.V., Baysal, O., Hacioglu, A.: Path planning for autonomous UAV via vibrational genetic algorithm. Aircraft Engineering and Aerospace Technology **79**(4), 352–359 (2007)
3. Yang, K., Sukkariéh, S.: 3D Smooth Path Planning for a UAV in Cluttered Natural Environments. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (2008)
4. Zhuoning, D., Rulinb, Z., Zongjia, C., Ruia, Z.: Study on UAV path planning approach based on fuzzy virtual force. Chin. J. Aeronaut. **23**(2010), 341–350 (2010)
5. Duan, H., Yu, Y., Zhang, X., Shao, S.: Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm. Simul. Model. Pract. Theory **18**(2010), 1104–1115 (2010)
6. Dong, Z., Chen, Z., Zhou, R., Zhang, R.: 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. Grøtli, E.I., Johansen, T.A.: Path planning for UAVs under communication constraints using SPLAT! and MILP. J Intell Robot Syst **2012**, 265–282 (2012)
8. Chamseddine, A., Zhang, Y., Rabbath, C.A., Join, C., Theilliol, D.: Flatness-Based Trajectory Planning/Replanning for a Quadrotor Unmanned Aerial Vehicle. IEEE Transactions On Aerospace And Electronic Systems **48**(4), 2012 (2012)
9. Wei, L., Zheng, Z., Kai-Yuan, C.: Bi-level programming based real-time path planning for unmanned aerial vehicles. Knowl.-Based Syst. **44**(2013), 34–47 (2013)
10. Roberge, V., Tarbouchi, M., Labonté, G.: Comparison of parallel genetic algorithm and particle swarm optimization for Real-Time UAV path planning. IEEE Trans. Indust. Inform. **9**(1), 2013 (2013)
11. Yan, F., Liu, Y., Xiao, J.Z.: Path planning in complex 3D environments using a probabilistic roadmap method. International Journal Of Automation And Computing **10**(6), 525–533 (2013)
12. Valente, J., Del Cerro, J., Barrientos, A.: David Sanz, Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. Comput. Electron. Agric. **99**(2013), 153–159 (2013)
13. Roberge, V., Tarbouchi, M., Allaire, F.: Parallel Hybrid Meta-heuristic on shared memory system for Real-Time uav path planning. International Journal of Computational Intelligence and Applications **13**(2), 1450008, 16 (2014)
14. Bircher, A., Kamel, M., Alexis, K., Burri, M., Oettershagen, P., Omari, S., Mantel, T., Siegwart, R.: Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. Auton Robot (2015)
15. Ortiz-Arroyo, D.: 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
16. Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., Siegwart, R.: Receding horizon path planning for 3D exploration and surface inspection. Auton Robot **42**(2018), 291–306 (2016)
17. Ingersoll, B., Ingersoll, K., DeFranco, P., Ningx, A.: UAV path-planning using bézier Curves and a receding horizon approach. AIAA Modeling and Simulation Technologies Conference (2016)
18. Radmanesh, M., Kumar, M.: 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. da Silva Arantes, M., Toledo, J., da Silva Arantes, C.F.M., Williams, B.C.: A hybrid multi-population genetic algorithm for UAV path planning, GECCO '16, July 20-24 Denver, CO, USA (2016)
20. Hayat, S., Yanmaz, E., Brown, T.X., Bettstetter, C.: 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. Zhang, T., Huo, X., Chen, S., Yang, B., Zhang, G.: 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. Zhang, G., Hsu, L.: A New Path Planning Algorithm Using a GNSS Localization Error Map for UAVs in an Urban Area. Journal of Intelligent & Robotic Systems (2019)
23. Papachristos, C., Mascarich, F., Khattak, S., Dang, T., Alexis, K.: Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning. Autonomous Robots. <https://doi.org/10.1007/s10514-019-09864-1> (2019)
24. J. Dhulkefl, E., Durdu, A.: Path Planning Algorithms for Unmanned Aerial Vehicles, International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456 (2019)
25. Liu, X., Du, X., Zhang, X., Zhu, Q., Guizani, M.: Evolution-algorithm-based unmanned aerial vehicles path planning in complex environment. Computers and Electrical Engineering **80**(2019), 106493 (2019)
26. Wai, R.J., Prasetya, A.S.: Adaptive neural network control and optimal path planning of UAV surveillance system with energy consumption prediction. IEEE Access **7**, 2019 (2019)
27. Freitas, H., Faiçal, B.S., Vinicius Cardoso e Silva, A., Ueyama, J.: 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. Qu, C., Gai, W., Zhong, M., Zhang, J.: A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. Applied Soft Computing Journal (2020). <https://doi.org/10.1016/j.asoc.2020.106099> (2020)
29. Choset, Howie, M.: Principles of robot motion: theory algorithms and implementations. MIT Press (2005)
30. Schøler, 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 (2012)
31. LaValle, S.M.: Planning algorithms. Cambridge University Press, Cambridge (2006)
32. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research **30**(7), 846–894 (2011)
33. Ahmad, Z., Ullah, F., Tran, C., Lee, S.: Efficient energy flight path planning algorithm using 3-D visibility roadmap for small unmanned aerial vehicle. International Journal of Aerospace Engineering (2017)
34. Carson, B.H.: Fuel efficiency of small aircraft. J. Aircr. **19**(6), 473–479 (1982)

35. Cavcar, M.: The international standard atmosphere (ISA), p. 30. Anadolu University, Turkey (2000)
36. Nachmani, G.: Minimum-energy flight paths for UAVs using mesoscale wind forecasts and approximate dynamic programming. DTIC Document (2007)
37. Wei, L., Zheng, Z., Kaiyuan, C.: Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval. *Chinese Journal of Aeronautics* **2013**,26(3), 646–660 (2013)
38. Zhang, Z., Scaramuzza, D.: Perception-aware Receding Horizon Navigation for MAVs, 2018 IEEE International Conference on Robotics and Automation (ICRA) May 21-25, 2018. Brisbane, Australia (2018)
39. Faria, M., Marín, R., Popovic, M., Maza, I., Viguria, A.: Efficient lazy theta* path planning over a sparse grid to explore large 3D volumes with a multirotor UAV. *Sensors* **2019**(19), 174 (2019)
40. Nash, A., Koenig, S., Tovey, C.: Lazy Theta*: Any-angle path planning and path length analysis in 3D. *Proceedings of the AAAI Conference on Artificial Intelligence* (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Giuseppe Aiello was born in Naples the 9th of January 1992. After graduating in Sports Science at the University of Cassino and Southern Latium in 2014, Giuseppe moved on Engineering areas obtaining a Bachelor's Degree in Computer Engineering at the University of Cassino and Southern Latium in 2018 and a Master's Degree in Mechatronic Engineering at the Polytechnic of Turin in 2020. He performed his researches on path planning for fixed-wing UAVs at the University of Denver during 2020. Since late 2020, Giuseppe works as a Control & Software Engineer in the Railway Industry.

Dr. Kimon P. Valavanis is currently John Evans Professor, D. F. Ritchie School of Engineering and Computer Science, University of Denver. He has been Guest Professor in the Faculty of Electrical Engineering and Computing, University of Zagreb; Visiting Professor at Politecnico di Torino, Dipartimento di Ingegneria Meccanica e Aerospaziale; and Professeur Invité, Université de Lorraine - Polytech Nancy. Dr. Valavanis' research interests focus on the areas of Unmanned Systems, Distributed Intelligence Systems, Robotics and Automation. He has published more than 450 book chapters, technical journal, and transaction papers, referred conference papers, invited papers and technical reports. He has authored, co-authored, and edited 19 books. He also holds several patents in unmanned aerial vehicles.

He was Editor-in-chief of the IEEE Robotics and Automation Magazine for ten years (1996–2005), and since 2006 he serves as Editor-in-chief of the Journal of Intelligent and Robotic Systems. He serves as Co-chair of the Aerial Robotics and Unmanned Aerial Vehicles Technical Committee since 2008. Dr. Valavanis has served as a Technical Expert of the NATO Science and Technology Organization (STO). He was appointed in 2021 to the NATO STO Technical Team of SAS-ET-EX on "Integration of Unmanned Systems into Operational Units" for the duration of the Program of Work. He is Fellow of the American Association for the Advancement of Science, Fellow of the U.K. Institute of Measurement and Control, IEEE Senior Member, and a Fulbright Scholar (*Senior Lecturing & Research Award*).

Alessandro Rizzo received the Laurea degree (summa cum laude) in computer engineering and the Ph.D. degree in automation and electronics engineering from the University of Catania, Italy, in 1996 and 2000, respectively. In 1998, he worked as a EURATOM Research Fellow with JET Joint Undertaking, Abingdon, U.K., researching on sensor validation and fault diagnosis for nuclear fusion experiments. In 2000 and 2001, he has worked as a Research Consultant at ST Microelectronics, Catania Site, Italy, and as an Industry Professor of robotics with the University of Messina, Italy. From 2002 to 2015, he was a tenured Assistant Professor with the Politecnico di Bari, Italy. Since 2012, he has been a Visiting Professor with the New York University Tandon School of Engineering, Brooklyn, NY, USA.

In November 2015, he joined Politecnico di Torino, where he is an Associate Professor in the Department of Electronics and Telecommunications and established the Complex Systems Laboratory. Dr. Rizzo is engaged in conducting and supervising research on cooperative robotics, complex networks and systems, modeling and control of nonlinear systems. He is the author of two books, two international patents, and about 200 papers on international journals and conference proceedings. He has been a recipient of the Award for the Best Application Paper at the IFAC world triennial conference in 2002 and of the Award for the Most Read Papers in Mathematics and Computers in Simulation (Elsevier) in 2009. He has also been a Distinguished Lecturer of the IEEE Nuclear and Plasma Science Society and one of the recipients of the 2019 and 2021 Amazon Research Awards in robotics.