# Multiobjective Path Planner for UAVs Based on Genetic Algorithms

**4 authors**, including:

Jesús Manuel de la Cruz
Complutense University of Madrid
**201** PUBLICATIONS   **2,367** CITATIONS

SEE PROFILE

Eva Besada
Complutense University of Madrid
**56** PUBLICATIONS   **462** CITATIONS

SEE PROFILE

Luis de la Torre Cubillo
National Distance Education University
**60** PUBLICATIONS   **496** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Easy JavaScript Simulations (EjsS) and EjsS extensions View project

4th Experiment@ International Conference 2017 - exp.at'17 View project

# MULTIOBJECTIVE PATH PLANNER FOR UAVS BASED ON GENETIC ALGORITHMS[*]

J.M. DE LA CRUZ-GARCÍA, E. BESADA-PORTAS, L. DE LA TORRE-CUBILLO, B. DE ANDRÉS-TORO

*Departamento de Arquitectura de Computadores y Automática,*
*Universidad Complutense de Madrid,*
*Av. Complutense s/n. Madrid, 28040, Spain*

This paper presents a path planner for Unmanned Air Vehicles (UAVs) based on Genetic Algorithms (GA) that obtains a feasible and optimal 3-D path for the UAV. It uses 9 different objective values which are calculated with a realistic model of the UAV and the environment and which are structured with 3 levels of priorities. Our planner works globally offline as well as locally online, which means that the algorithm can recalculate parts of the generated path in order to avoid unexpected risks. Finally, the effectiveness of the solutions given by this planner has been successfully tested against a simulator that contains the complete model of the UAV and the environment.

**Keywords:** genetic algorithms, multiobjective planning, UAVs

## 1. Introduction

UAVs are the logical evolution in aerospace technology, due to the advances in micro-controllers, new control theory and optimization techniques. The actual importance of this subject can be observed in the great number of publications that are currently appearing in the fields of automatic path planning and automatic control of vehicles. Currently, these are the two basic problems that must be solved: the route generation and the control of the vehicle. This paper is about the first problem.

There is already a lot of work done to solve this problem with different algorithms such as A* [1] or MILP [2], although in this paper, we will show the advances done with GAs and present our approach.

GAs have already been used by some authors to find a near optimal path for UAVs. In [3] and [4] the problem is formulated as finding the waypoints that define a spline curve that the UAV will follow and which is feasible and optimal

according with some criteria. Nikolos et al. [3] designed a GA to find trajectories for UAVs for offline and online navigation, using 4 objectives (for offline planning) that were added to transform the problem onto a monobjective one. Mittal and Deb [4] extended the work in [3] implementing a multiobjective evaluation, which allowed constraints considerations.

Our work uses a multiobjective GA that introduces some new ideas besides constraining the problem with the properties of a F-16 model for the UAV. First, we use a relative polar coordinates codification for the waypoints of the path, which accelerates the convergence of the algorithm. Second, the user can force the path to pass through several specific points and some of them can be defined as refuelling points. Third, instead of using a mathematical function that simulates hills and valleys, our terrain is a real model of any part of the world. Fourth, UAVs' velocities are considered and used on the fuel consumption calculation. Fifth, flight prohibited zones can be defined. Sixth, 9 different objectives and constraints are considered in the fitness function and combined with a pareto multiobjective evaluation function.

## 2. Algorithm Codification

Our individual is a list of 3D points (waypoints) which determine the spline curve that constitutes the solution of the problem: a 3D path. So, our algorithm searches for the best list of waypoints which define a feasible and optimal spline trajectory for the UAV.

Each waypoint is defined in the 3D space by 3 coordinates (such as 'x', 'y' and 'z'). The codification of 'z' is simple because it can be limited by the valid flying altitudes. For 'x' and 'y', instead of using absolute Cartesians coordinates which produce a big search space, we select a relative coordinate codification that forces the algorithm to have a predefined order of waypoints. This is positive because that order already exists and if it is not imposed by the codification the algorithm has to find it. So, the selected codification let us dramatically reduce the search space and computation time.

To determine the 'x' and 'y' position of the waypoint, we use the relative polar coordinates 'r' and '$\theta$'. Given the start point, the algorithm will search from the first waypoint in a radius r≤80 km. Values for $\theta$ will be limited to a certain interval given by [$\theta_{min}$, $\theta_{max}$], which depends on the UAV manoeuvrability. The first $\theta$ angle is the angle formed between the initial vector direction and the vector direction defined by the start point and the first waypoint. For any other waypoint, this angle is measured according with the schema in figure 1.
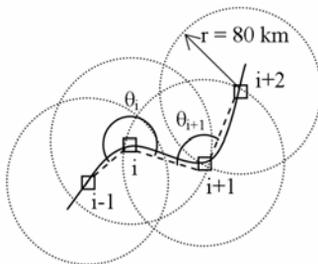
Figure 1. Relative polar coordinates for 'x' and 'y'. The values of θ are measured following the meaning clockwise respect to the (i-1,i) vector direction. This way, θmax = 2π - θmin. The solid line represents the spline curve (which is the path of the UAV) obtained from the waypoints.

The main drawback of our codification (a list of 'r', 'θ' and 'z' tuples) is that the algorithm has to include a constraint that prevents the path to get outside the map limits, but the GA satisfies this condition in only a few generations.

## 3. Fitness Function

There are several objectives that the UAV must satisfy completely (constraints) and others that have to be minimized (optimization indexes). They are measured over the spline curve and not directly over the waypoints. The evaluation method used to rank the trajectories according with their objective values is the generic Fonseca multiobjective method [5] based on goals, priorities and Pareto sets, where each objective value can be limited and placed in a selected priority level.

In our problem, we make use of 9 objectives: 6 constraints (higher priority objectives) and 3 optimization indexes (lower priority ones). More objectives can be easily included, due to the generic method use to rank the trajectories. Although all the objective values are obtained with simplified models that consider the real properties of the system, the whole simulator could be used to measure them at the expenses of increasing significantly the computation time.

- **Constraint Conditions:** A feasible path that the UAV can follow must fulfil 6 constraints. The first five (1-5) measure the number of times each of them is violated, while the last (6) is an upper limit that musn't been exceeded. All the constraints are placed in the first (highest) priority level.
1. *Terrain.* The path mustn't get under the terrain or collide with a mountain. So, if any point of the spline curve is under it, this objective value is incremented.
2. *Turning radius.* The UAV can't follow any path where 3 consecutives points form a turning radius smaller than the minimum permitted by the physical characteristics of the UAV.

3. *Map limits.* The UAV must stay inside the limits of our known map. This means that every point of the trajectory must fulfill this condition.

4. *Maximum climbing and diving slope.* UAVs can't reach a greater slope than the one that is given by their characteristics (which can be different when climbing or diving).

5. *Flight prohibited zones (or Not Flying Zones, NFZs).* There can be certain zones where the UAV must not enter: high risk areas, unknown zones, etc. So, the algorithm also checks if any point of the curve is inside a NFZs.

6. *Fuel.* UAVs have a limited quantity of fuel and they have to reach their goal or a refuelling point before consuming it. We use a model that estimates the fuel consumption in three different cases: a) when the UAV is flying horizontally, b) when it is flying with maximum slope and c) when it is flying with maximum turning angle. For intermediate cases the fuel consumption is calculated as a weighted mean of the 3 previous cases.

- **Optimal Path for the Mission:** The optimality of the path is determined by the following 3 optimization indexes. The first two objectives (7 and 8) are in the second priority level while the last one (9) is in the third.

7. *Minimum length path.* Shorter paths are better than longer ones because they require less time of flight. The cost function for this objective is just the length of the spline curve defined by the waypoints.

8. *Minimum probability of kill (or Pk).* This function depends on the model used for the Air Defence Units (ADUs). Based on many simulations we have obtained a function of the probability of killing an UAV dependent on the distance and the relative altitude between the UAV and the ADU. The penalization will be the accumulated Pk along the whole path.

9. *Minimum flight altitude.* UAVs flying at low altitude can benefit from terrain mask effect (that will help them to avoid radars) and reduce the fuel consumption. This objective is the difference between the 'z' coordinates of the points of trajectory and the corresponding altitude of the terrain.

## 4. Experimental Results

A great number of route planifications in different scenarios have been done with our algorithm and the solutions have been tested, with positive results, with a simulator that uses a complex model for the UAV and the ADUs. We present two examples that reflect important characteristics of the planner.

- *First example,* which shows the ability of the algorithm to find existing no risky paths between the Start point and Goal (see figure 2). This scenario contains 3 ADUs and 2 NFZs. The big circles mark the maximum distance of detection of each ADU while the small ones enclose the zones where Pk>0. NFZs are represented as rectangular grey zones. Note that there is a little corridor (of 5 kms) where Pk=0 and that the algorithm has been able to

find it even though the Pk and the lenght path objectives are in the same priority level. Figure 3 shows that the altitude is also being minimized; the trajectory is always below the Goal altitude, which is higher than the Start point. As the path selected by the algorithm is feasible and safely optimal, all the tests carried out in the simulator have succeeded to reach the goal.
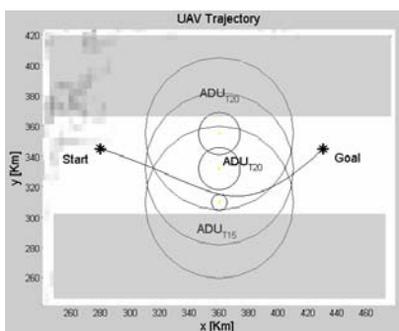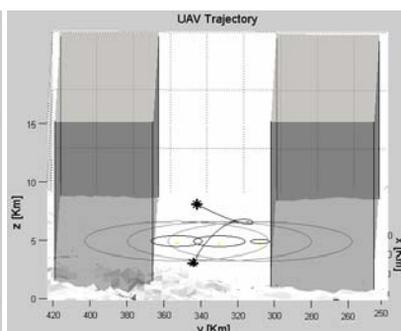


Figure 2. Eagle eye view          Figure 3. Isometric 3D view

- *Second example,* which illustrates how the algorithm works with pop-ups, obligatory crossing points and roundtrip paths. There are 3 obligatory crossing points (WP$_2$, WP$_3$, WP$_4$), 1 ADU (located at the northeast part of the map), 1 NFZ and 1 pop-up ADU (near the Start point, at the southwest part of the map). The continuous line is the original generated path (figure 4), while the dotted line (better observable in figure 5) is the local new path calculated by the algorithm when the pop-up appears. This local plan evades the pop-up with success, not only according with the optimization indexes of the trajectory, but also during the simulations inside the complex system.
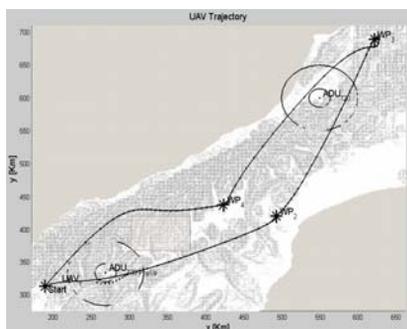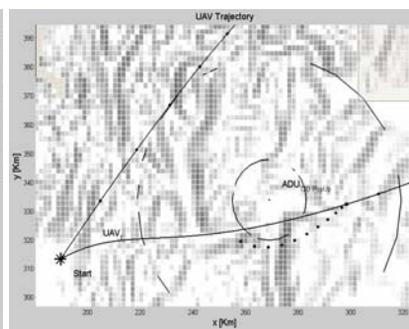


Figure 4. Eagle eye view          Figure 5. Eagle view near the pop-up

## 5.  Conclusions

Relative polar coordinates are a good choice to codify solutions because they accelerate the convergence of the algorithm without loss of generality while the new constraint condition required (map limits) is easily fulfilled.

The Fonseca evaluation method allows finding solutions that fulfil all the constraint conditions (so the physical limitations on the movement of the UAV are completely respected) while the other objectives are optimized.

Adding new objectives and constraints (such as the minimum altitude flight or the fuel consumption) doesn't prevent the GA from finding good solutions and so, the planner could be prepared to face more general scenarios and flight missions. Moreover, the simplified realistic model of the UAV, used in the fitness function works great (as we checked with the simulator) so the possibility of using the GA to find paths for a F-16 real flight mission could be explored.

Our current work also studies the simultaneously path generation for several UAVs, like in [6]. This can be made having several populations (one for each UAV) that evolves independently, except for the coordination objectives.

## References

1.  Karen Irene Trovato, *'A\* Planning in Discrete Configuration Spaces of Autonomous Systems'*. PhD thesis, Amsterdam University, 1996.
2.  José J. Ruz, Orlando Arévalo, Jesús M. de la Cruz and Gonzalo Pajares, *'Using MILP for UAVs Trajectory Optimization under Radar Detection Risk'*, 11th IEEE International Conference on Emerging Technologies and Factory Automation, Prague, September 2006. IEEE 2006, pp 957-960.
3.  Ioannis K. Nikolos, Kimon P. Valavanis, Nikos C. Tsourveloudis and Anargyros N. Kostaras, 'Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation', *IEEE Transactions on Systems, Man, and Cybernetics* – Part B : Cybernetics, Vol 33, NO. 6, December 2003.
4.  S. Mittal and K. Deb, 'Three-Dimensional Offline Path Planning for UAVs Using Multiobjective Evolutionary Algorithms'. *Kangal Report* N0. 2004008
5.  Fonseca C.M., Fleming P.J., *'Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms- Part I: Unificied Formulation'*. Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans. Vol 28, nº1, January 1998, pp 26-37.
6.  Changwen Zheng, Lei Li, Fanjiang Xu, Fuchun Sun and Mingyue Ding, 'Evolutionary Route Planner for Unmanned Air Vehicles', *IEEE Transaction on Robotics*, Vol. 21, NO. 4 August 2005.