Επιχειρησιακή Έρευνα / Operational Research. An International Journal. Vol.5, No.3 (2005), pp.487-502

Coordinated UAV Path Planning Using Differential Evolution

Athina N. Brintaki, Ioannis K. Nikolos

Department of Production Engineering and Management,
Technical University of Crete, University Campus,
GR-73100, Chania, Greece.
E-mail: jnikolo@dpem.tuc.gr

Abstract

A Differential Evolution based framework is utilized to design an offline path planner for Unmanned Aerial Vehicles (UAVs) coordinated navigation in known static maritime environments. Considering the problem of having a number of UAVs starting from different known initial locations, the issue is to produce 2-D trajectories, formed by successive way-points, with a desirable velocity distribution along each trajectory, aiming at reaching a predetermined target location, while ensuring collision avoidance either with the environmental obstacles or with the UAVs and satisfying specific route and coordination constraints and objectives. The constraints are imposed in order to maximize the probabilities of UAVs survival and mission accomplishment.

Keywords: UAVs, coordinated path planning, Differential Evolution

1. Introduction

This work has been motivated by the challenge to develop and implement an offline path planner for Unmanned Aerial Vehicles (UAVs) coordinated navigation and collision avoidance in known static maritime environments, characterized by the existence of a number of islands with short distances between them.

UAVs are increasingly being used in real-world applications, as they provide high maneuverability, low risk and weight and cost savings, compared with manned aircraft. Planning an UAV's flight path is one of the various problems in autonomous UAV deployment; the corresponding path planner should provide feasible flight paths that satisfy specific time, space, recourse and availability constraints. Path planning is actually a multi-objective multi-constraint optimization problem, in most cases very complex and computationally demanding [Mettler et. Al. (2003)]. Several methods that produce trajectories for such vehicles in known, unknown or partially known environments have been proposed [Nikolos et. Al. (2003)], [Nikolos et. Al. (2001)], [Sasiadek et. Al. (2000)], [Kuwata et. Al. (2004)]. However, the problem complexity increases when multiple UAVs should be used, e.g. in missions with requirements beyond the capabilities of a single UAV, or when the probability of mission accomplishment increases with the number of involved UAVs. Several approaches have been proposed in the literature for UAVs coordinated route planning. In [Beard et. Al. (2002)], [McLain et. Al. (2000)], a Voronoi-diagram approach is presented, which deals with the problem of simultaneous arrival of multiple UAVs at their targets, using threat-avoiding trajectories. In [Schouwenaars et. Al. (2004)] a framework is presented for decentralized trajectory planning of multiple autonomous aircraft, using a receding horizon strategy based on mixed integer linear programming. In [Flint et. Al. (2002)] the problem of generating near-optimal trajectories, in order multiple UAVs to cooperatively search for targets, is addressed, while a dynamic programming method is presented as a solution to the problem.

During the past few years, it has been shown by many researchers that Evolutionary Algorithms (EAs) is a viable candidate to solve path planning problems effectively and provide feasible solutions within an affordable time without demanding excessive computer power. EAs have been successively used for the solution of the path-finding problem in ground based, sea surface, aerial, or underwater vehicles navigation [Nikolos et. Al. (2003)], [Nikolos et. Al. (2001)], [Michalewicz (1999)], [Smierzchalski (1999)], [Smierzchalski et. Al. (2000)], [Sugihara et. Al. (1997a)], [Sugihara et. Al. (1997b)]. In this work, Differential Evolution (DE) is used to produce the routes for multiple UAVs in known static maritime environments. Considering a number of UAVs at different known initial locations, the issue is to produce 2-D trajectories, formed by successive way-points, with a desirable velocity distribution along each trajectory, that reach a common target, under specific coordination and route constraints.

The rest of the paper is organized as follows: Section 2 describes the problem formulation, including assumptions, objectives, constraints and cost function definition. Section 3 summarizes EA fundamentals along with the basic features of Differential Evolution. Experimental results are presented in Section 4, followed by discussion and conclusions in Section 5.

2. Problem Formulation

2.1 Assumptions, Objectives, and Constraints

The aim of this work is to develop a coordinated path planner for navigation and collision avoidance of a team of autonomous UAVs in maritime environments. The environments considered in this work are known and static, characterized by the existence of a number of islands with short distances between them. The flight height is assumed to be almost constant, close to the sea-level, and this is the reason why the problem is considered as 2-D. Considering the problem of having *N* UAVs starting from different known initial locations, the issue is to produce *N* 2-D trajectories, formed by successive way-points, with a desirable velocity distribution along each trajectory, aiming at reaching a predetermined target location, while ensuring collision avoidance either with the environmental obstacles or with the UAVs and satisfying specific route and coordination objectives and constraints. The environment has known characteristics and flight restrictions, acquired via 3-D GIS based generated maps or otherwise.

UAVs are assumed to be equipped with a set of on-board sensors, including GPS (Global Positioning System) / DGPS (Differential GPS), INS (Inertial Navigation System) and gyroscopes, through which they can sense their positions and flight directions. Each vehicle is assumed to be a point; its actual size is taken into account by equivalent obstacle – ground growing.

The general constraint of the problem is the collision avoidance between UAVs and the ground obstacles.

The route constraints are the following:

- Predefined initial and target coordinates for each UAV.
- Predefined initial and final velocity magnitudes for each UAV.
- Predefined minimum and maximum UAV velocity magnitudes, during their flights.

Additionally, a single route objective is imposed:

- Minimum path lengths, in order to maximize the effective range of each vehicle.

The three route constraints are problem-defined (initial and target coordinates), or they depend on the flight envelop of each UAV (velocity magnitudes). All three of them are explicitly taken into account by the optimization algorithm. The route objective is implicitly handled by the algorithm, through the cost function definition.

Besides route constraints and objective, coordination-relative constraints and objectives exist among UAVs which form a team to accomplish a joint mission. The coordination objectives used in this work are listed below:

- Arrival at target with minimum time intervals. Each UAV should arrive to the target, using a different path and a different approach vector, but the time of arrival for all UAVs should be as close as possible.
- Approach the target from different directions. In addition to the spatial separation between different flight corridors, the target approach from different flight directions maximizes the probability of mission accomplishment. For this reason, all the angles between successive approaching directions should be as equal as possible, in order to assure an almost uniform distribution of UAVs around the target.

The single coordination constraint is defined as:

- Keep a safety distance between UAVs.

Each UAV has to keep a safety distance from the rest UAVs, during the whole flight. The safety distance is addressed primarily in order to ensure collision avoidance between UAVs. The second reason is to ensure the maximum spatial separation between the corresponding flight corridors, which, for some missions, increases the probability of survival. Coordination constraint and objectives are implicitly handled by the algorithm, through the cost function definition.

Each path is constructed by straight line segments, which connect the successive way points. In order to construct each segment, its length seg_length_{kj}, and its direction seg_angle_{kj} are used as design variables ($k=1,\dots, n-1, j=1,..,N$, N being the number of UAVs, while $n+1$ is the number of way points in each path, the same for all paths). Design variables seg_angle_{kj} are defined as the difference between the direction (in deg.) of the current segment and the previous one. For the first line segment of each path seg_angle_{kj} is measured from x-axis. Additionally, the UAV's velocities $c_{k,j}$ at each way point are used as design variables, except for the starting and target points.

Using seg_length_{kj} and seg_angle_{kj} the coordinates of each way point $x_{k,j}$ and $y_{k,j}$ can be easily calculated. The use of seg_length_{kj} and seg_angle_{kj} as design variables instead of $x_{k,j}$ and $y_{k,j}$ was adopted for two reasons. The first reason is the fact that abrupt turns between successive flight paths can be easily avoided by explicitly imposing short lower and upper bounds for the seg_angle_{kj} design variables. The second reason is that using the proposed design variables a better convergence was achieved, compared to the case with the way points' coordinates as design variables. The latter observation is a consequence of the shortening of the solutions space, using the proposed formulation.

For starting ($k=0$) and target points ($k=n$) the user predefines their coordinates, along with the corresponding velocity magnitudes. Additionally, the velocity of each UAV is assumed to vary linearly between successive way points. The lower and upper boundaries of each independent design variable are predefined by the user. Velocity boundaries depend on the flight envelope of each UAV. For the first segment of each path seg_angle_{1j} upper and lower boundaries can be selected such as to define an initial flight direction. Additionally, by selecting lower and upper boundaries for the rest of seg_angle_{kj} variables close to 0 degrees (for example -30^o to 30^o), abrupt turns can be avoided.

2.2 Cost Function Formulation

The problem of computing the optimum path for each UAV is formulated as a minimization one, and the corresponding cost function is formulated such as to take into account the general constraint of collision avoidance between UAVs and the ground, the single route objective (minimum path lengths), the two coordination objectives and the single coordination constraint.

The cost function to be minimized is formulated as the weighted sum of five different terms

$$f = \sum_{i=1}^{5} w_i f_i \tag{1}$$

where w_i are the weights and f_i are the corresponding terms described below.

Term f_1 is the sum of the non-dimensional lengths of all N flight paths and corresponds to the single route objective:

$$f_1 = \sum_{j=1}^{N} l_j \tag{2}$$

where l_j is the non-dimensional length of the j^{th} path, given as

$$l_j = \frac{L_j}{\sqrt{\left(x_{target} - x_{0,j}\right)^2 + \left(y_{target} - y_{0,j}\right)^2}} - 1 \tag{3}$$

and

$$L_j = \sum_{k=1}^{n} \sqrt{\left(x_{k,j} - x_{k-1,j}\right)^2 + \left(y_{k,j} - y_{k-1,j}\right)^2} \tag{4}$$

In (3) and (4) L_j is the length of the j^{th} path, x_{target}, y_{target} are the coordinates of the target point, $x_{0,j}$, $y_{0,j}$ are the coordinates of the j^{th} starting point, $x_{k,j}$, $y_{k,j}$ are the coordinates of the k^{th} way point of the j^{th} path, and $n+1$ is the number of way points in each path, including the starting ($k=0$) and target ($k=n$) points. In (3), for the calculation of the non-dimensional length l_j, the distance between the starting and target points is subtracted, in order to obtain zero f_1 value for straight line paths. For all N paths the number of way points is the same, determined by the user. Term f_1 is used in order to minimize the flight path lengths.

Term f_2 is a penalty term, which was designed in order to materialize the general constraint of collision avoidance between UAVs and the ground. All n path segments of all N flight paths are checked whether or not pass through each one of the M ground obstacles. Discrete points are taken along each path line (using the simulation procedure described later) and they are checked whether or not they lie inside an obstacle. If this is true for a discrete point of the path line, a constant penalty is added to term f_2. Consequently, term f_2 is proportional to the number of discrete points that lie inside obstacles. Additionally, for each path line, a high penalty is added in case that even one discrete point of the corresponding path lies inside an obstacle.

Term f_3 was designed in order to take into account the second coordination objective, i.e. the target approach from different directions. For each flight path the opposite to the flight direction azimuth angle of the last path segment is calculated as (Figure1):

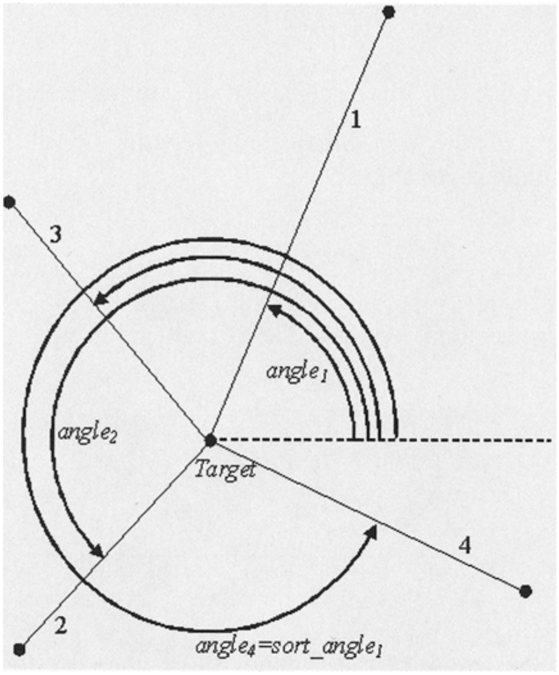

Figure 1. *Definition of azimuth angles, calculated for the last path segment of each flight path*

$$\begin{aligned} angle_j &= \arctan(\Delta y/\Delta x),\ if\ \ \Delta y \geq 0\ and\ \Delta x \geq 0 \\ angle_j &= 2\pi - \arctan(\Delta y/\Delta x),\ if\ \ \Delta y < 0\ and\ \Delta x \geq 0 \\ angle_j &= \pi + \arctan(\Delta y/\Delta x),\ if\ \ \Delta x < 0 \\ \Delta y &= y_{n\text{-}1,j} - y_{n,j},\quad \Delta x = x_{n\text{-}1,j} - x_{n,j} \end{aligned} \qquad \textbf{(5)}$$

All calculated azimuth angles $angle_j$, $(j=1,\ N)$ are sorted in a descending order and stored as variables $sort\_angle_j$. An additional variable $sort\_angle_{N+1}$ is calculated as:

$$sort\_angle_{N+1} = sort\_angle_1 - 2\pi \qquad \textbf{(6)}$$

Subsequently, the deference between two successive $sort\_angle_j$ is calculated as

$$\Delta sort\_angle_j = sort\_angle_j - sort\_angle_{j+1}, \quad j = 1,\dots,N \tag{7}$$

where $\Delta sort\_angle_j$ is the angle between two successive flight path segments, connected to target point (Figure2).

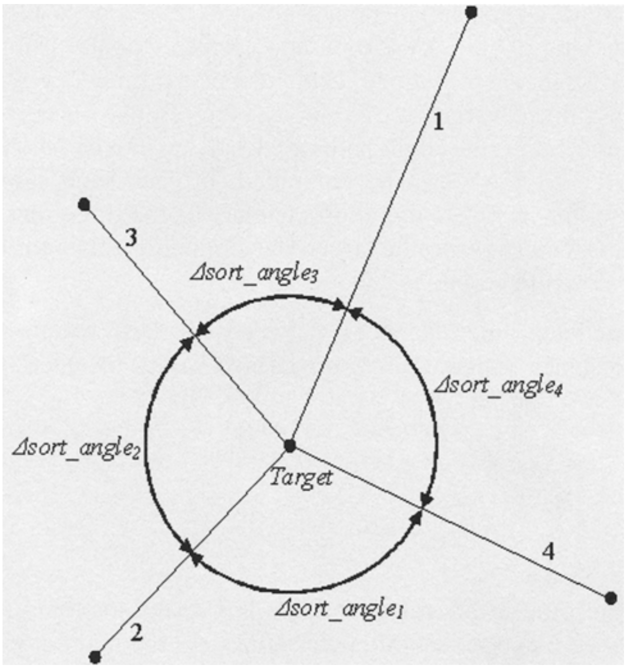

Figure 2. *Definition of Δsort_angle$_j$, calculated for the last path segment of each flight path, according to (7)*

We define *opt_angle* as:

$$opt\_angle = 2\pi / N \tag{8}$$

The variable *opt_angle* denotes the optimum angle between successive flight paths as UAVs are approaching the target in order to have uniform distribution of UAVs around the target. Term f_3 is then calculated as:

$$f_3 = \frac{\sum_{j=1}^{N} \left| opt\_angle - \Delta sort\_angle_j \right|}{ref\_angle} \tag{9}$$

In (9), *ref_angle* is a small reference angle which is used to provide a non-dimensional form of f_3 and takes a value equal to $\pi/20$.

Term f_4 is relevant to the single coordination constraint (keep a safety distance between UAVs), while term f_5 is relevant to the first coordination objective (arrival at target with minimum time intervals). For their calculation, a flight simulation is needed. Each candidate solution is defined by the corresponding design variables. Then the coordinates of all way points are computed, while the coordinates and the velocities at the starting and target points are predefined by the user. Assuming a simultaneous launching of all UAVs from their corresponding starting points at $t=0$, a simulation of their flights is performed, using a constant time step Δt, the same for all UAVs. Assuming a linear variation of velocity between two successive way points k and $k+1$ of the same path j, the flight path of each UAV can be reconstructed, and the actual position of each UAV can be computed, in each time step. When a UAV arrives at a way point its flight direction changes to the direction of the next path segment. When a UAV arrives to the target, the simulation stops for this UAV and its time of arrival is stored in variable t_curr_j.

During each time step, the distances between all UAVs current positions are calculated. If a distance is less than a predefined safety distance d_{safe}, a penalty is added to term f_4.

Term f_5 is calculated as

$$f_5 = \sum_{j=1}^{N} \left(t\_\max - t\_curr_j\right) / t\_\max \tag{10}$$

where t_max is the time of arrival of the last UAV (the maximum flight duration). Weights w_i in (1) were experimentally determined. As the main objective is to obtain feasible flight paths, weights were determined in a way that term $w_2 f_2$ dominates the rest.

3. The Optimization Method

3.1 Fundamentals of Evolutionary Algorithms

The path planning problem described above is formulated as a minimization problem of the cost function f. An Evolutionary Algorithm (EA) was adopted for solving the problem; the main reason was the fact that EAs are more robust than other search methods in global optimization problems. EAs [Michalewicz (1999)], [Holland (1992)], [Goldberg (1989)], are characterized by a remarkable balance between exploitation of the best solutions and exploration of the search space. Additionally, they may be easily tailored to the specific application of interest, taking into account the special characteristics of the problem under consideration, and they are easily parallelized. Despite their advantages, all the population-based search algorithms, such as EAs, require excessive CPU time, due to the large number of evaluations of candidate solutions.

The natural selection process is simulated in EAs, using a number (population) of individuals (solutions to the problem) to evolve through certain procedures. Each individual is represented through a string of numbers (bit strings, integers or floating point numbers), in a similar way with chromosomes in nature. Each individual's quality is represented by a fitness function tailored to the problem to be optimized.

Classical EAs use binary coding for the representation of the genotype. However, floating point coding moves EAs closer to the problem space, allowing the operators to be more problem specific [Michalewicz (1999)]. Additionally, two points that are close in the physical space are also close in the representation space and vice versa. With this type of encoding directed search techniques gain physical representation and they are easily applicable.

In this work, Differential Evolution (DE), a recently proposed EA, is used [Storn et. Al. (1995)], [Storn (1995)]. DE is an extremely simple to implement EA, which has demonstrated better convergence performance than other EAs. It can handle continuous, discrete and integer variables, and multiple constraints. DE has been demonstrated to be one of the most promising novel EAs, in terms of efficiency, effectiveness and robustness. Its main characteristics are described below.

3.2 Differential Evolution

Let us consider an optimization problem formulated as

$$\min_{X} = f(X) \tag{11}$$

where X is a vector of nx1 parameters which forms a chromosome and f the cost function to be minimized. In each generation G a population of N_p candidate solutions undergoes specific operations, so a candidate solution can be designated as

$$X_{i,G}, \quad i = 1, \ldots, N_p, \quad G = 1, \ldots, mgens \tag{12}$$

where *mgens* is the maximum number of generations, and N_p does not change during the optimization process. Parameter vector X in our case contains floating point parameters.

The DE algorithm starts by generating randomly, with uniform probability, the initial chromosome population, with their genes taking values inside the predefined constrained space. The crucial idea behind DE is a new mutation scheme for generating trial parameter vectors, by adding the weighted difference vector between two population members to a third member, which is called the donor. The mutation operation is applied for all population members. For each individual, the mutation process begins by randomly selecting three individuals among the current population, which form a triplet. In this triplet one member is randomly taken to be the donor, while the other two members are taken to produce a perturbation to the donor. In this way, the i^{th} perturbed individual is generated as

$$V_{i,G+1} = X_{r3,G} + F\left(X_{r1,G} - X_{r2,G}\right) \tag{13}$$

where

$$r1,\ r2,\ r3\ \in \left\{1,\ldots,N_p\right\},\ r1 \neq r2 \neq r3 \neq i \tag{14}$$

are randomly selected, among the candidate solutions of the current population. The scaling parameter *F*, introduced in [Storn et. Al. (1995)], is a control parameter of DE algorithm set by the user, taking values in the interval:

$$F \in [0,\ 1+] \tag{15}$$

F is constant along the evolution procedure and controls the amplification of the perturbation added to the donor. By giving larger values to *F*, a higher exploration capability is obtained. The perturbed individual $V_{i,G+1}$ and the initial population member $X_{i,G}$ are then subjected to the crossover operation, which generates the intermediate population of trial vectors $U_{i,G+1}$. If,

$$\begin{aligned} X_{i,G} &= \left(x_{1,i,G},\ldots,x_{n,i,G}\right)^T \\ V_{i,G+1} &= \left(v_{1,i,G+1},\ldots,v_{n,i,G+1}\right)^T \\ U_{i,G+1} &= \left(u_{1,i,G+1},\ldots,u_{n,i,G+1}\right)^T \end{aligned} \tag{16}$$

then

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \textit{if}\ \ rand_j \leq C_r \vee j = k \\ x_{j,i,G} & \textit{otherwise} \end{cases} \tag{17}$$

where $j = 1,\ldots,n$ and $k \in \{1,\ldots,n\}$ is a random index, chosen once for all N_p members of the population. The crossover parameter $C_r \in [0,1]$ is the second control parameter, set by the user. The individuals that will form the next generation are selected between the current population and the corresponding trial vectors, according to the following rule:

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \textit{if}\ \ f\left(U_{i,G+1}\right) \leq f\left(X_{i,G}\right) \\ X_{i,G} & \textit{otherwise} \end{cases} \tag{18}$$

A recently proposed scheme [Hui-Yuan et. Al. (2003)] to determine the donor for mutation operation was introduced, for accelerating the convergence rate. In this scheme, donor is randomly selected (with uniform distribution) from the region within the "hyper-triangle", formed by the three members of the triplet

$$donor = \sum_{i=1}^{3}\left(\lambda_i \Big/ \sum_{j=1}^{3}\lambda_j\right) X_{ri,G}, \quad \lambda_j = rand_j[0,1] \tag{19}$$

where $rand_j[0,1]$ denotes a uniformly distributed value within the range [0,1]. With this scheme the donor comprises the local information of all members of the triplet, providing a better starting-point for the mutation operation that result in a better distribution of the trial-vectors. As it is reported in [Hui-Yuan et. Al. (2003)], the modified donor scheme accelerated the DE convergence rate, without sacrificing the solution precision or robustness of the DE algorithm.

The random number generation (with uniform probability) is based on the algorithm developed by [Marse et. Al. (1983)], which computes the remainder of divisions involving integers that are longer than 32 bits, using 32-bit (including the sign bit) words. The corresponding algorithm, using an initial seed, produces a new seed and a random number. In each different operation inside the DE algorithm that requires a random number generation, a different sequence of random numbers is produced, by using a different initial seed for each operation and a separate storage of the corresponding produced seeds. By using specific initial seeds for each operation, it is ensured that the different sequences differ by 100,000 numbers.

4. Experimental Results

4.1 The Solid Boundary Representation

The terrain where UAVs fly is most generally represented by a meshed 3-D surface produced using mathematical functions of the form:

$$\begin{gathered} z(x,y) = \sin(y+a) + b\cdot\sin(x) + c\cdot\cos\left(d\cdot\sqrt{y^2+x^2}\right) + \\ e\cdot\cos(y) + f\cdot\sin\left(f\cdot\sqrt{y^2+x^2}\right) + g\cdot\cos(y) \end{gathered} \tag{20}$$

where *a, b, c, d, e, f, g* are constants experimentally defined, in order to produce a surface simulating a maritime environment with islands close to each other (as shown in Figure3).

A graphical interface has been developed for the visualization of the terrain surface, along with the path line [Nikolos et. Al. (2003)]. The corresponding interface deals with different terrains produced either artificially or based on real geographical data, providing an easy verification of the feasibility and the quality of each solution. The path-planning algorithm considers the boundary surface as a group of quadratic mesh nodes with known coordinates.

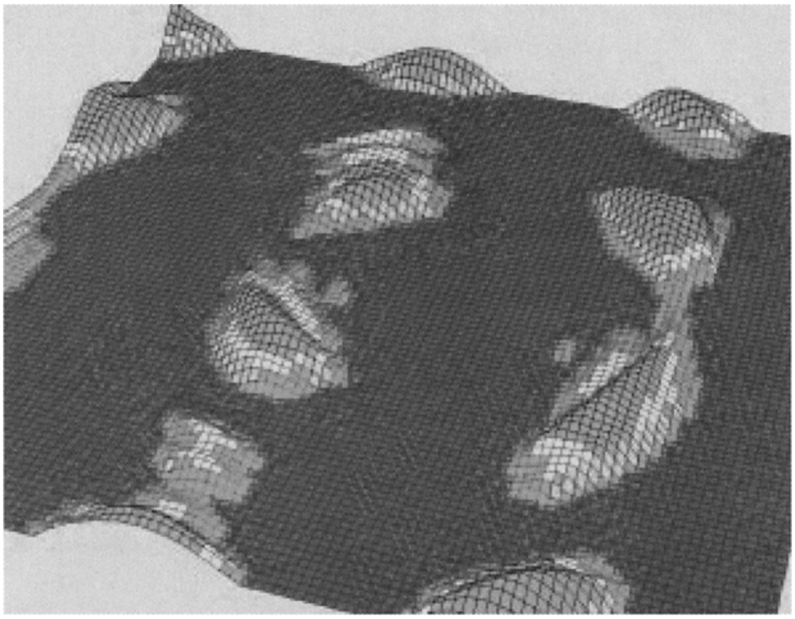

Figure 3. *The first Test Case*

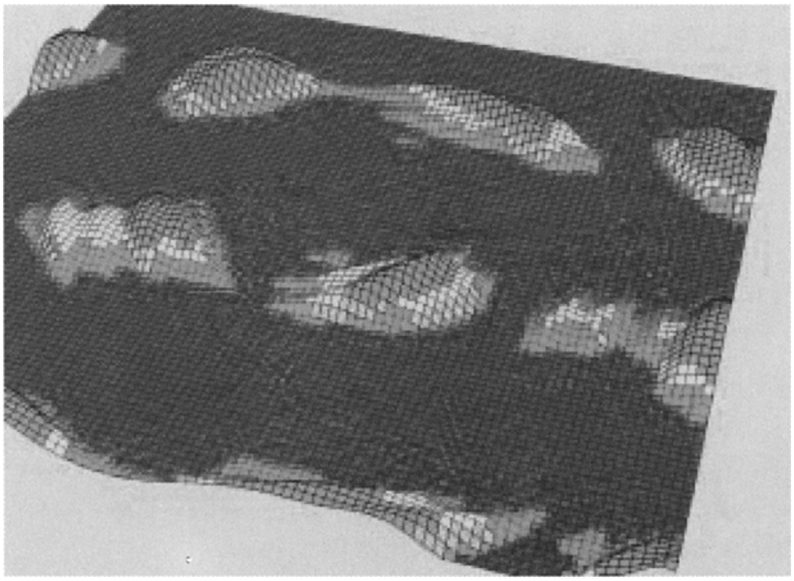

Figure 4. *The second Test Case*

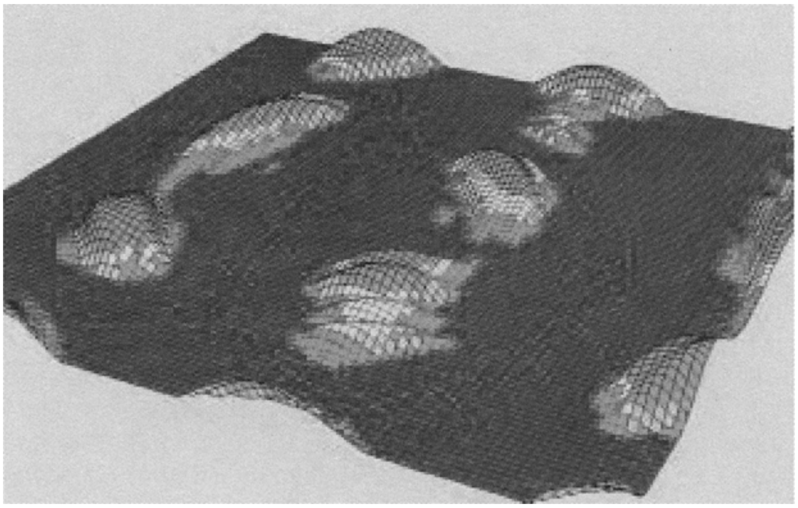

Figure 5. *The third Test Case*

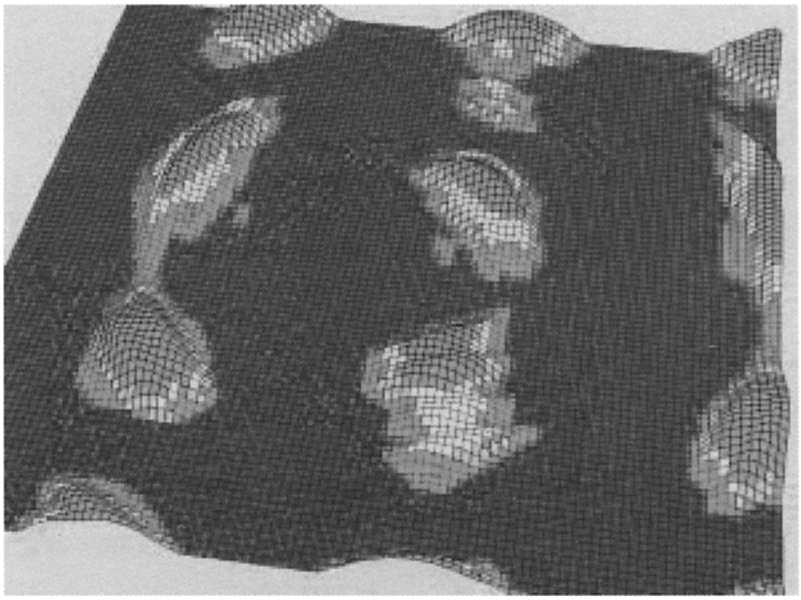

Figure 6. *The fourth Test Case*

4.2 Simulation Results

The same environment was used for all the test cases considered, with different starting and target points. The (experimentally optimized) settings of the Differential Evolution algorithm are as follows: population size = 50, F = 1.05, C_r = 0.85. The algorithm was defined to terminate after 200 generations, although feasible solutions can be reached in less than 30 iterations. With 200 generations and a population size equal to 50, 10,000 evaluations of the cost function are performed before the algorithm stops. In the test cases presented here, the free-to-move way points for each path were taking values equal to 3, resulting in a total number of way points equal to 5 for each path (along with the fixed starting and target points). For 3 different paths (corresponding to 3 UAVs) and 3 free-to-move way points for each path, a total number of 27 design variables are needed (*seg_length*$_{kj}$, *seg_angle*$_{kj}$ and $c_{k,j}$, for each path *j* and each way point *k*).

Figures 3 to 6 present simulation results for four different test cases. In all test cases considered, target points were positioned relatively close to solid boundaries, in order to test the ability of the algorithm to provide feasible path-lines with uniform distribution of UAVs around the target. For all test cases safety distance d_{safe} was set equal to 12.5% of the length of each side of the rectangular terrain. For all test cases, term f_4 of the cost function converged to zero, corresponding to no violation of the safety distance constraint. For the first three cases the final path segments are uniformly distributed around the target. The less uniform distribution of the final flight path segments, presented in Fig. 6, is due to the fact that the target was placed very close to the adjacent island.

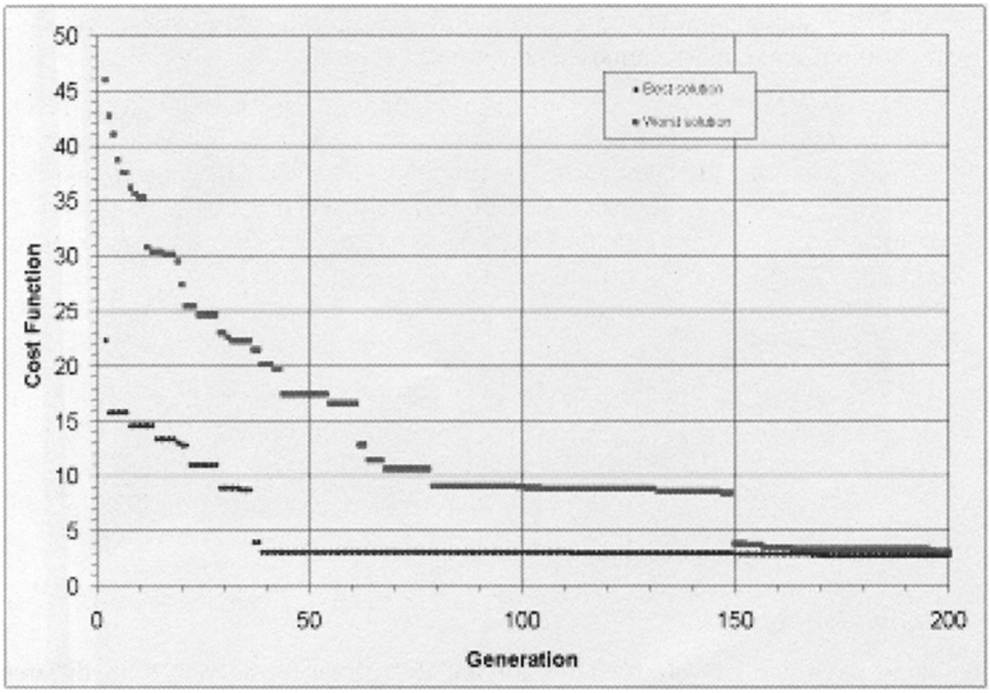

Figure 7. *Convergence history for the fourth Test Case*

Concerning the time intervals between the first and the last arrival to the target, for all the test cases considered, this time interval was kept less than 1% of the flight duration (0.84% for the 1st case, 0.64% for the 2nd case, 0.55% for the 3rd case and 0.43% for the 4th case).

The convergence history of the Differential Evolution algorithm is presented in figure 7, for the fourth test case. The cost functions of the best and the worst individual of each generation are plotted against the generation number. The individuals with a cost function value below 10 correspond to feasible solutions; less than 30 generations are needed in order to produce feasible solutions.

5. Conclusions

In this work an offline path planner for Unmanned Aerial Vehicles (UAVs) coordinated navigation and collision avoidance in known static maritime environments was presented. The problem is actually a multi-objective, multi-constraint optimization one, but it was formulated as a single-objective optimization problem, by defining a single cost function as the weighted sum of five different terms. Those terms correspond to different route and coordination objectives and constraints. The path planner was tested in a simulated environment, and the simulation results demonstrated the ability of the algorithm to produce near optimal paths without violating the imposed constraints.

The DE algorithm proved to be effective in finding feasible path lines under the forced constraints within an acceptable time period. The easy implementation of the various constraints and objectives of the problem was a valuable characteristic of DE algorithm. Moreover, a feasible solution could be reached within a small number of generations, while the rest of the generations were used in order to optimize the solution, according to the rest of the criteria.

References

Beard R.W., McLain T.W., Goodrich M.A. and Anderson E.P. (2002). Coordinated Target Assignment and Intercept for Unmanned Air Vehicles. IEEE Transactions on Robotics and Automation, vol. 18(6), pp. 911-922, Dec. 2002.

Flint M., Polycarpou M., and Fernandez-Gaucherand E. (2002). Cooperative Control for Multiple Autonomous UAV's Searching for Targets. Proceedings of the 41st IEEE Conference on Decision and Control.

Goldberg D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Holland J.H. (1992). Adaptation in Natural and Artificial Systems. The MIT Press.

Hui-Yuan F., Lampinen J., Dulikravich G.S. (2003). Improvements to Mutation Donor Formulation of Differential Evolution. Evolutionary Methods for Design, Optimization and Control, Applications to Industrial and Societal Problems, Proc. of EUROGEN 2003, CIMNE, Barcelona.

Kuwata Y., and How J. (2004). Three Dimensional Receding Horizon Control for UAVs. Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA-2004-5144.

Marse K. and Roberts S.D. (1983). Implementing a portable FORTRAN uniform (0,1) generator. Simulation, 41:135.

McLain T.W. and Beard R.W. (2000). Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles. AIAA-2000-4369.

Mettler B., Schouwenaars T., How J., Paunicka J., and Feron E. (2003). Autonomous UAV guidance build-up: Flight-test Demonstration and evaluation plan. Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA-2003-5744.

Michalewicz Z. (1999). Genetic Algorithms + Data Structures = Evolution Programs. Springer Publications.

Nikolos I.K., Valavanis K.P., Tsourveloudis N.C., Kostaras A (2003). Evolutionary Algorithm based offline / online path planner for UAV navigation. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 33(6), pp. 898-912, Dec. 2003.

Nikolos I.K., Tsourveloudis N. and Valavanis K.P. (2001). Evolutionary Algorithm Based 3-D Path Planner for UAV Navigation. CD-ROM Proceedings of the 9th Mediterranean Conference on Control and Automation, Dubrovnik, Croatia.

Sasiadek J.Z. and Duleba I. (2000). 3D Local Trajectory Planner for UAV. Journal of Intelligent and Robotic Systems, vol. 29, pp. 191-210, 2000.

Schouwenaars T., How J., and Feron E. (2004). Decentralized Cooperative Trajectory Planning of multiple aircraft with hard safety guarantees. Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA-2004-5141.

Smierzchalski R. (1999). Evolutionary trajectory planning of ships in navigation traffic areas. Journal of Marine Science and Technology, Vol. 4, pp. 1-6.

Smierzchalski R. and Michalewicz Z. (2000). Modelling of ship trajectory in collision situations by an evolutionary algorithm. IEEE Transactions on Evolutionary Computation, Vol. 4, pp. 227-241.

Storn R. and Price K. (1995). DE - a Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Space. ICSI, Technical Report TR-95-012.

Storn R. (1995). Differential Evolution Design of an IIR-Filter with Requirements for Magnitude and Group Delay. ICSI, Technical Report TR-95-026.

Sugihara K. and Smith J. (1997a). Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot. Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, California, pp. 138-143.

Sugihara K. and Yuh J. (1997b). GA—based motion planning for underwater robotic vehicles. UUST-10, Durham, NH.