3D UAV trajectory planning using evolutionary algorithms: A comparison study

M. Bagherian
mbagherian@guilan.ac.ir
Applied Math Department
Faculty of Mathematical Science
University of Guilan
Rasht
Iran

A. Alos
ahm_syr@yahoo.com

ABSTRACT

This paper focuses on the three dimensional flight path planning for an unmanned aerial vehicle (UAV) on a low altitude terrain following\terrain avoidance mission. The UAV trajectory planning problem is to compute an optimal or near-optimal trajectory for a UAV to do its mission objectives in a surviving penetration through the hostile enemy environment, considering the shape of the earth and the kinematics constraints of the UAV. Using the three dimensional terrain information, three dimensional flight path from a starting point to an end point, minimising a cost function and regarding the kinematics constraints of the UAV is calculated. The geographic information of the earth shape and enemy locations is generated using digital terrain model (DTM) and geographic information system (GIS), and is displayed in a 3D environment. Using 3D-maps containing the geographic data accompanied by DTM, and GIS, the problem is modelled by deriving the motion equations of the UAV. Two heuristic algorithms are proposed for this problem: genetic and particle swarm algorithms. Genetic and particle swarm algorithms are general purposes

Paper No. 4356. Manuscript received 8 February 2015, accepted 7 May 2015.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

algorithms, because they can solve a wide range of problems, so they have to be adjusted to solve the trajectory planning problem. To test and compare the paths obtained from these algorithms, a software program is built using GIS tools and the programming languages C# and MATLAB.

NOMENCLATURE

DTM	digital terrain model
DEM	data elevation model
$Height(x,y)$	the value of any earth point elevation
γ	climb angle
ψ	turn rate
$J_{distance}$	distance cost
J_{length}	path length cost
J_{height}	height cost
$J_{obstacle}$	obstacle cost
DTED	digital terrain elevation data
Δv	speed rate
$\Delta\gamma$	climb angle rate
$\Delta\psi$	heading angle rate
$f^j(x^j, y^j, z^j)$	fitness function

1.0 INTRODUCTION

It is very important to plan a trajectory for a UAV before sending it to do its mission; this process has received considerable research attention, because following the planned trajectory maximises the likelihood of mission succession of a flying vehicle, Chen[1]. The objective of the UAV trajectory planning is to find the optimum flight path that maximises survivability while satisfying appropriate flight path constraints. Past approaches to this problem have usually formulated the problem as a matter of numeric cost minimisation to be solved by methods like dynamic programming, (Belman[2]), numerical procedures, (Betts and Huffman[3]) or geometric algorithms like A^* search (Rippel *et al*[4]). A^*-based algorithms which are variants of the shortest path algorithm, have difficulties with route constraints. Moreover these algorithms are strongly based on the cost map which should be produced and stored; and the cost map production is a very time consuming task. In dynamic programming a local cost is assigned to each link of the grid which spans the area of operation. This assumes that the cost of flying over an area is independent of the path by which the UAV reached the area, which is not always true. Both dynamic programming and A^* search techniques suffer from their relatively slow execution. Some researchers have formulated the optimal path planning problem as an optimal control problem which is equivalent to a two-point boundary value problem, (Betts and Huffman[3]). As discussed in Betts and Huffman[5] this approach is problematic, since the adjoin equations are nonlinear and obtaining them is difficult for complex dynamics. Moreover, convergence of the corresponding iterative solution procedure is often very sensitive to the accuracy of the initial guess of the adjoin variables; and there is no physical interpretation of the adjoin variables to provide an intuition into what constitutes a good initial guess. Many authors have investigated alternative approaches to the solution of the optimal control problem: Enright and Conway[6], Hall[7] and Lu[8]. However their approaches suffer from long convergence time. The interested reader is refereed to Hwang and Ahuja[9] and Latombe[10]

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

for more discussion about different route planning approaches. Chen *et al* [1] presented a good approach of the GA algorithm for two dimensional trajectory planning with a variable-length chromosome; however it was only suitable for 2D areas, and did not handle terrain following (TF)/ threat avoidance (TA) missions. So, in this paper, this algorithm is extended to be applicable for three dimensional trajectories and accept more different changes in genes parameters values like (speed rate, climbing angle rate, and heading angle rate). In addition by collecting the information about some important DTM concepts and motion equations a model is constructed for the problem. Then two evolutionary algorithms are proposed to solve the problem: genetic algorithm (GA), and particle swarm optimisation (PSO) algorithm which are general purpose algorithms. Since they can be used to solve a large domain of problems, they have to be adjusted to be applicable to trajectory planning problem.

Without loss of generality, the path planning problem considered in this paper can be described as follows.

Given a start point at (x_0, y_0, z_0); a target point at (x_T, y_T, z_T); and a set of obstacles located at (x^i_0, y^i_0, z^i_0), where $i = \{1, 2, \ldots .N\}$, respectively, find a near-optimal UAV trajectory $(x^j, y^j, z^j) = \{(x_k, y_k, z_k)\}$ defined at times $\{t_k, k = 0, 1 \ldots N\}$, which advances toward the target point, considering the UAV dynamics. So the path should be generated as a sequence of speed rate, climb angle rate, and heading angle rate at discrete times $\{t_k, k = 0, 1 \ldots N\}$, respectively, while taking into account the physical constraints of the climb angle, γ and turn rate, ψ. The rest of the paper is organised as follows. Section 2 describes the terrain representation. Section 3 models the UAV. Section 4 describes the UAV path planning. Section 5 and 6 dedicates to GA and PSO algorithms respectively. Computational results are presented in Section 7 and Section 8 ends the paper with conclusion.

2.0 TERRAIN REPRESENTATION

A topographic† 3D-map can be defined as a cartographic‡ representation of a landscape section in a perspective view, combined with topographic information that is defined in a legend. The topographic information of a topographical 3D-map includes all terrain characteristics (height, exposition, slope) and other topographic elements such as lakes, rivers, settlement areas as well as other infrastructures, road and railway networks, land use and vegetation patterns, (Haeberling[11]). Basically 3D-maps are topographic information plus DTM. The digital terrain model (DTM) is simply a statistical representation of the continuous surface of the ground by a large number of selected points with known x, y, z co-ordinates in an arbitrary co-ordinate system. In computer language the earth surface or earth shape will be digital terrain model (DTM), or data elevation model (DEM).

This DEM or DTM should be accurate and real, and geo-referenced. In this paper, WGS84 is used for the geographical co-ordinate system (longitude, latitude), and UTM for the projected co-ordinate system. The DEM will be a geographic TIFF raster image file representing the area as a rectangular grid of cells according to a suitable resolution, each cell will have a height value and its location is defined by two values (x, y), the resolution of the DEM is defined by the cell size. In this paper the resolution of some TIFF images is represented by:

$$\text{cellsize} = 85{\cdot}9890611,\ 85{\cdot}9890611 \text{ metres.}$$

† Topographic: is the description of earth's surface shape and features, especially their depiction on the map
‡ Cartographic: is the study and practice of making maps

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

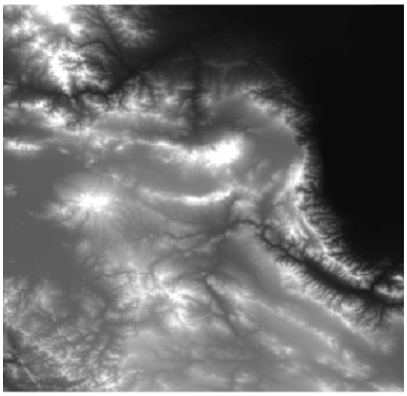


Figure 1. Terrain representation.

Figure 1 shows a graphical representation of the DEM.

The DEM is a grey scale raster, the value of each pixel is an integer bounded from 0 to 255. The value of any earth point elevation is calculated using minimum height value and maximum height value by the following simple equation:

$$Height(x,y) = (cellvalue(x,y)*(h_{max} - h_{min}))/255 + h_{min} \quad \ldots (1)$$

Here x, y are the cell column order, and row order in the raster grid respectively.

3.0 UAV MODEL

In this paper the flight model equations and UAV physical constraints are similar to Zardashti and Bagherian [12]. Cost function components are like Chen[1], Anargyros[13] and Ioannis and Kimon[14]. Based on Call[15] an algorithm is proposed to avoid obstacles and threats.

A simplified kinematics model of a UAV flying in a three-dimensional airspace is of interest. So the UAV is considered as a point in a 3D space, and its translational and angular states at time t are defined as:

$$X(t) = (x(t), y(t), z(t), \gamma(t), \psi(t))^T \; G$$

$$G = R^3 \times (\gamma_{min}, \gamma_{max}) \times (-\pi, +\pi)$$

The co-ordinates x, y and z are taken with respect to an inertial reference frame. The flight speed is $V(t)$. The state equations could be written as:

$$\dot{X}(t) = V(t)\text{Cos}\,\gamma(t)\text{Cos}\,\psi(t) \quad \ldots (2)$$

$$\dot{Y}(t) = V(t)\text{Cos}\,\gamma(t)\text{Sin}\,\psi(t) \quad \ldots (3)$$

$$\dot{Z}(t) = V(t)\text{Sin}\,\gamma(t) \quad \ldots (4)$$

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

Also it will be useful to add some physical constraints on the climb angle and turn rate γ, ψ, respectively; for example:

$$\gamma_{min} \leq \gamma \leq \gamma_{max} \quad \ldots (5)$$

$$\dot{\psi} \leq \dot{\psi}_{max} \quad \ldots (6)$$

4.0 UAV PATH PLANNING

The low altitude flight and penetration mission can be divided into two main steps in mission execution. The first step is the mission preparation, in which the task is to collect information about the mission area, prepare the airborne electronic terrain map, and plan an optimal reference flight route so as to improve the mission survivability of the UAV. The second step is the mission execution, where the airborne computer calculates the optimal flight trajectory based on the planned flight route and local information of the terrain and threats, then follows the trajectory to finish the flight mission. In this step, because the information detected and processed by the airborne equipment is limited, a planned flight route obtained before the mission is necessary for the trajectory optimisation so as to ensure the optimality of the mission. Therefore the flight route planning is very important for improvement of mission survivability and successability of the UAV Li Qing *et al*[16].

As mentioned, the path planning problem considered in this paper can be described as follows.

Suppose we have a start point at (x_0, y_0, z_0); a target point at (x_T, y_T, z_T); a set of obstacles located at (x^i_0, y^i_0, z^i_0), where $i = \{1, 2, \ldots N\}$ respectively, and we have to find a near-optimal UAV trajectory $(x^j, y^j, z^j) = \{(x_k, y_k, z_k)\}$ defined at times $k = \{1, 2, \ldots, N\}$, which advances toward the target point.

However, the previous model cannot be used directly while considering the UAV dynamics, and the path should be generated from a sequence of speed rate, climb angle rate, and heading angle rate at discrete times $\{t_k, k = 0, 1 \ldots N\}$ respectively, taking into account the physical constraints of the climb angle and turn rate (γ, ($\dot{\psi}$)), as mentioned above.

The path changes should be decoded to generate a corresponding UAV trajectory (x^j, y^j, z^j) for the flight. Using Equations (2), (3), and (4), and considering discrete times, the motion of the UAV is described by the following equations:

$$v(k+1) = v(k) + \Delta v \quad \ldots (7)$$

$$\gamma(k+1) = \gamma(k) + \Delta\gamma \quad \ldots (8)$$

$$\psi(k+1) = \psi(k) + \Delta\psi \quad \ldots (9)$$

$$x(k+1) = x(k) + v(k+1)\text{Cos}\,\gamma(k+1)\text{Cos}\,\psi(k+1) \quad \ldots (10)$$

$$y(k+1) = y(k) + v(k+1)\text{Cos}\,\gamma(k+1)\text{Sin}\,\psi(k+1) \quad \ldots (11)$$

$$z(k+1) = z(k) + v(k+1)\text{Sin}\,\gamma(k+1) \quad \ldots (12)$$

Where v is the UAV speed, γ is the climb angle, ψ is the heading angle, and x, y, z are the inertial co-ordinates of the UAV position.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

Table 1
Path changes

Row	Angle rates								
v	0	2	10	6	3	5	3	1	0
$\Delta\gamma$	0	–2	0	2	0	1	0	–1	0
$\Delta\psi$	–2	0	4	0	2	1	0	3	–1

where v, $\Delta\gamma$, $\Delta\psi$ are the speed, climb and heading angle rates, respectively

This discrete UAV model is based on the assumption that there exist inner and outer loop navigation control laws, which enable the UAV to track an acceptable and safe trajectory as long as speed, climb angle and heading angle rates satisfy the physical constraints.

The process of finding the desired path must minimise a cost function $J(x^j, y^j, z^j)$, Chen[1].

5.0 GA APPROACH

Briefly, the GA algorithm is initiated with an initial set of randomly generated chromosomes called the initial population. The genes of each chromosome are a sequence of numerical values. In this paper each chromosome or individual represents a UAV trajectory constrained by UAV dynamics. The population will be changed repeatedly in each generation by genetic operations (crossover, mutation, selection, insertion, deletion); the changed elements (chromosomes) of the population will be selected according to some fitness function. The target of this process is minimising the fitness function as far as required, or finding the chromosome which has the near-minimal fitness value finally. This chromosome is called the near-optimal solution.

A detailed explanation of GA algorithm could be found in references Back[17], Goldberg[18] and Melanie[19].

As mentioned reference Chen[1] presented a good approach of the GA algorithm for two dimensional problems with a variable-length chromosome; however it was only suitable for 2D areas, and did not handle TF/TA problems. So in this paper this algorithm is extended to be applicable for three dimensional trajectories, and accept more different changes in genes parameters values like (speed rate, climbing angle rate, and heading angle rate). Finally a cost function component is used to handle the TF/TA problem, considering the UAV physical constraints like maximum speed, maximum climbing angle … etc.

Table 1 shows an example of the speed, climb angle and turn angle rates sequenced in a random length chromosome.

5.1 Cost function

Usually, the cost function $J(x^j, y^j, z^j)$ is a weighted positive scalar function, which must reflect all the forces that conspire to derail the intensions of the UAV, and consists of several components including distance cost, path length cost, obstacle and the height costs.

5.1.1 Distance cost $J_{distance}$

The distance cost is the distance between the terminal point on a trajectory, and the target point. Suppose the termination time of the flight is t_N, which is a free parameter, because in this paper a variable-length sequence is used to present a flight path. So $J_{distance}$ can be computed as:

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

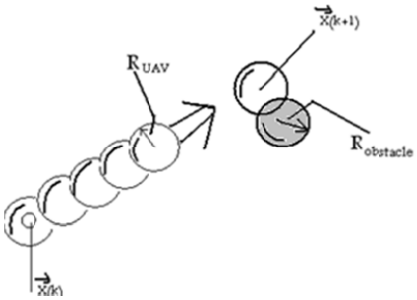


Figure 2. Obstacle cost calculating.

$$J_{distance} = Distance(\vec{X}(t_N), TARGET) \quad \ldots (13)$$

Minimising this cost value insures nearing the target point.

5.1.2 Path length cost J_{length}

This cost is used to find shorter paths. It can be expressed as

$$J_{length} = \sum_{k=0}^{N-1} Distance(\vec{X}(t_{k+1}), \vec{X}(t_k)) \quad \ldots (14)$$

Where N is the number of the sample intervals.

Minimising this cost value insures to near to the shortest flying path.

5.1.3 Height cost J_{height}

This cost is used to avoid hitting the ground, and assures that the UAV flies in low altitudes. To calculate this component f_3, we need the vertical distance h_i between any point in the trajectory and the surface of the Earth. Assuming x, y, z are the co-ordinates of the ith point p_i of the trajectory then:

$$h_i = p_i(z) - terrain(x,y)$$

Every point in the trajectory must reach a vertical distance, whose value is bounded between the minimum and maximum safe height. Thus the cost f_3 is calculated using the following equations:

$$J_{height} = \sum_{i=1}^{c} cost_i$$

$$cost_i = \begin{cases} C: & \text{if}(h_i < \text{Minimum height}) \text{or } (h_i > \text{Maximum height}) \\ 0: & \text{otherwise} \end{cases}$$

Here C is a suitable constant value.

5.1.4 Obstacle cost $J_{obstacle}$

The UAV is considered as a moving ball of radius, R_UAV moving along a particular path segment from time t_k to $t_{(k+1)}$, as shown in Fig. 2.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

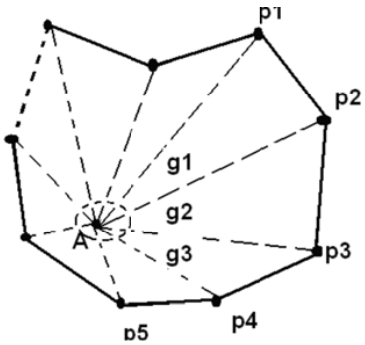


Figure 3. Calculating if the point is inside the polygon

This component $J_{obstacle}$ is calculated using the following expressions:

$$\boldsymbol{J_{obstacle}} = \sum_{k=0}^{N-1} \sum_{i=1}^{c} cost_{ki} \quad \ldots (15)$$

$$cost_{ki} = \begin{cases} D: & if \;\; p_k \; is \;\; inside \; (Espot_i) \\ 0: & otherwise \end{cases} \quad \ldots (16)$$

Where c is the obstacles count, N is trajectory points count, D is a suitable constant value, $Espot_i$ is the ith obstacle area.

In this paper the obstacle areas are considered as a randomly shaped convex or concave polygon. Consider the polygon represented in Fig. 3.

Let point A be a point included in the trajectory, and the obstacle polygon is constructed in a way that the points $p_1, p_2, p_3, \ldots p_{(i)}$ are its vertices. It is needed to know whether the point A is inside the obstacle area or not. To solve this problem; the sum of the angles formed from the points p_i, A, $p_{(i+1)}$:i:(1..polygon order) is calculated, if the sum equals 360° then the point A is inside the obstacle area, otherwise not.

$$A \in inside\ polygon \Leftrightarrow \sum_i \hat{g}_i = 360^{\circ} \quad \ldots (17)$$

where $\hat{g}_i = angle(p_i, A, p_{i+1})$

5.2 Proposed GA for UAV path planning

The next sub-sections describe the proposed GA algorithm for trajectory planning; including genetic representation, chromosome decoding, fitness function, and GA operators.

5.2.1 Genetic representation

Each chromosome consists of three rows of different integers, representing the speed rate, climb angle rate, and heading angle rate, at discrete times $\{t_k, k = 0, 1 \ldots N\}$ respectively, taking into account the physical constraints of maximum speed, the climb and turn rate γ, ψ, as mentioned previously. Table 2 shows an example of the changes contained in a random chromosome with random length, Pellazar and Miles[20].

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

Table 2
Genetic representation

Row	Genetic representation								
Δv	–5	–3	2	0	1	2	–3	–3	0
$\Delta\gamma$	0	–2	0	2	0	1	0	–1	0
$\Delta\psi$	–2	0	4	0	2	1	0	3	–1

where Δv, $\Delta\gamma$, $\Delta\psi$ are the speed rate, climb angle rate, and heading angle rate, respectively

5.2.2 Chromosome decoding

Now the genes of each chromosome should be decoded to generate a corresponding UAV trajectory (x^j, y^j, z^j) for the flight; as it is used to calculate the fitness function, and showing the result graphically. Equations (7) to (12) are used to decode and encode the chromosomes.

5.2.3 Fitness function

The fitness function evaluates the goodness of each chromosome. It must accurately measure the quality of the chromosomes in each generation, and in our problem it should calculate the total cost of a given trajectory, see Anargyros *et al*[13], Ioannis *et al*[14], Li Qing *et al*[16] and Chen[1]. This fitness function needs to be optimised, to get an optimal trajectory. Thus the fitness function $f^j(x^j, y^j, z^j)$ used here is the inverse of the weighted sum of the four previous costs:

$$f^j(x^j, y^j, z^j) = \frac{1}{\sum_{i=1}^{4} w_i J_i(x^j, y^j, z^j)} \quad \ldots (18)$$

where

$$\sum_{i=1}^{4} J_i(x^j, y^j, z^j) = J_{distance} + J_{length} + J_{height} + J_{obstacles} \quad \ldots (19)$$

In which (w_i) are the weights corresponding to each cost.

5.2.4 Genetic operators

Selection: This operator gives the high-quality chromosome a better chance to reproduce and get copied to the next generation. In this paper we use the weighted roulette wheel selection, which is based on the individual's selection probability (relative fitness). Thus, high-fitness individuals are likely but not guaranteed to be reproduced (Chen[1] and Pellazar[20]).

Crossover: The crossover operation allows two individuals to exchange information by swapping some part of their representation with other individuals. This creates new individuals that may or may not perform better than the parent individuals. The choice of which individuals to crossover and at what point is done randomly, and this is what gives GA much of its power, Pellazar[20]. In this paper we used one-point crossover. Figure 4 shows an example of the crossover procedure, Chen[1] and Pellazar[20].

Mutation: The random mutation operator is used to prevent the loss of information that occurs when there are many nearly-optimal individuals that are missing an important value in their genes. This operator adds a random factor to the individual members, without affecting the rest of the population. The individuals can 'jump' out of local minima and come closer to the optimal value.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

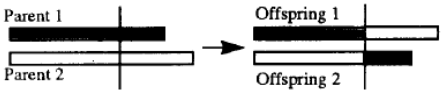


Figure 4. Example of the crossover procedure.

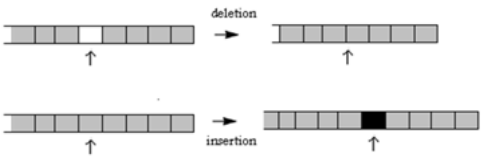


Figure 5. The insertion and deletion procedures.

Mutation must be used carefully, since it often causes near-optimal individuals to perform worse. The mutation rate is sometimes decreased linearly during the simulation, Pellazar[20].

Insertion and Deletion: these operators are necessary to implement variable-length chromosomes. They are added to the mutation function, where they apply to some of the candidate parents. The insertion operator inserts a gene into the selected chromosome, while the deletion operator deletes a gene from the candidate chromosome. Figure 5 shows two examples of these operators.

6.0 PSO APPROACH

Particle swarm optimisation (PSO) was originally designed and introduced by Eberhart and Kennedy[21]. The PSO is a population based search algorithm based on the simulation of the social behaviour of birds, bees or a school of fishes. Every bird recalls its flying experience which is the shortest distance between food and itself. Birds communicate their flying experience with each other, and then these behaviours lead all birds into the location of food, so it is called swarm intelligence, (Leandro[22], Chun-Yao and Yi-Xing[23], and Shi[24]). Each individual within the swarm is represented by a vector in multidimensional search space. This vector has also one assigned vector which determines the next movement of the particle and is called the velocity vector. Each particle updates its velocity based on current velocity and the best position it has explored so far; and also based on the global best position explored by swarm. The PSO process then is iterated a fixed number of times or until a minimum error is achieved based on a desired performance index Lazinica[25].

Particle swarm optimisation is similar to a genetic algorithm in that the system is initialized with a population of random solutions. It is unlike a genetic algorithm, however, in that each potential solution is also assigned a randomised velocity, and the potential solutions, called particles, are then flown through hyperspace.

References 23, 24 and 25 used the PSO algorithm to solve the trajectory planning problem without considering the kinematic constraints of the vehicle. Roberge *et al*[26] used GA and PSO algorithms for real-time UAV path planning and obtained quasi-optimal trajectories for fixed wing UAVs. Qiang[27] proposed an improved PSO algorithm for 3D path planning in a concentric spherical co-ordinate system. The generated trajectory will have the form of a variable-length line that advances slowly towards the target point, avoiding obstacles and enemy spots.

PSO algorithm resembles genetic algorithm, except it does not do genetic operations such as crossover and mutation. The particles move randomly till they find the target, but through their movement, the algorithm focuses on the particles with the least cost values.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

6.1 Representing particles

Each particle will be the UAV mass centre itself in one of its states, its location and orientation is defined by six parameters: x, y, z, v, γ, ψ and all the particles start from the point ($x_0, y_0, z_0, v_0, \gamma_0, \psi_0$), and fly in the virtual space according to motion Equations (7) to (12). In every time step (Δt) PSO makes each particle move ahead one step, stores the state of the particle (location, orientation) in an array like (x, y, z, v, γ, ψ), and calculates the cost value of the current location. Eventually PSO will store a set of arrays of locations, each array will be a potential UAV trajectory, but PSO will focus only on the best trajectory with the least cost value. PSO will stop whenever one of the trajectories reaches a location relatively acceptable to be near the target point and its cost is the least value.

6.2 The cost value

The cost value is calculated to show the goodness of the particle's trajectory at each step. The Cost function $f^j(x^j, y^j, z^j)$ can be considered as:

$$f^j(x^j, y^j, z^j) = \sum_{i=1}^{4} w_i J_i(x^j, y^j, z^j) \quad \ldots (20)$$

where:

$$\sum_{i=1}^{4} J_i(x^j, y^j, z^j) = J_{distance} + J_{length} + J_{height} + J_{obstacles} \quad \ldots (21)$$

and w_i are the weights corresponding to each cost.

7.0 EXPERIMENTAL RESULTS

Using GIS tools, MATLAB environment (m files), and C# programming language, the performance of the two algorithms are compared.

The DEM over which the trajectory is planned, is represented by a TIFF raster projected using (UTM -zone 38N) and the geoid WGS84 and covers an area nearly(450,407 × 541,342)m^2. The elevation ranges between 74 and 4,788 metres above the sea level. The maximum speed of the UAV is 300 metres/seconds. The climb angle ranges between45°, and 45°. The initial speed is 0 metres/seconds. The initial climb angle is 45°. Assuming that the 0°angle direction points towards the East; the initial direction angle will be 270°. The step time is 30 seconds, i.e. every 30 seconds the trajectory planning algorithm registers and adds a trajectory point. The flight height of the trajectory point should range between 100 and 600 metres.

Figure 6 shows four obstacle zones – the polygon areas filled with the red colour – that the UAV has to avoid flying near.

The distance between the start and target points is nearly 471,731 metres.

7.1 Testing GA

The initial population is chosen to be 20 genes, and the maximum number of the population is 30 genes; increasing these values is possible but it will cause the algorithm to run very slowly, especially if the distance between the start and the target point is very large.

The result of ten executions of the genetic algorithm is shown in Table 3.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

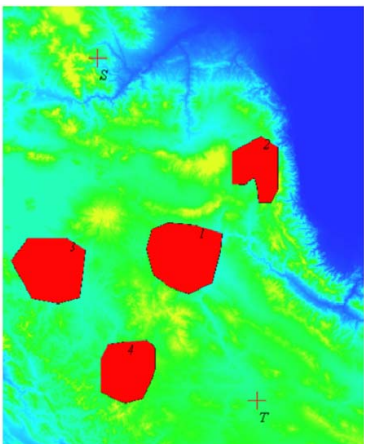


Figure 6. Obstacle areas.

Table 3
Testing GA

Name	Points Count	Length	Calc Time (m)	Time to reach target (Sec)	Cost (Sec)
GA	59	487,950	63·8	1,740	928,141·5
GA1	58	490,530	63	1,710	992,703,401·7
GA2	61	509,730	65·9	1,800	121,180,590·8
GA3	57	482,370	62·8	1,680	70,864,453·9
GA4	57	475,710	61·6	1,680	59,163,507·7
GA5	58	487,530	62·6	1,710	99,093,020·4
GA6	58	486,450	62·9	1,710	131,297,755·7
GA7	56	474,600	60·8	1,650	94,164,936
GA8	56	478,140	61·7	1,650	79,444,360·8

It can be seen that in every execution a totally new trajectory is created, this is acceptable because GA depends on stochastic operations. The best trajectory is GA which has the least cost, even though it does not reach the target in the shortest period of time. Since the cost function does not depend only on the shortest path, but it is calculated using other parameters such as (How much the trajectory points are far from the obstacle areas, or how much their heights are in the range between the minimum and the maximum value of height …), for example trajectory GA7 has the shortest length, but not the least cost, see Fig. 7.

Note that trajectory GA7 is too near to the obstacle zone 1, this way the cost will be larger.

Figure 8 shows a 3D representation of the two trajectories:

7.2 Testing PSO

By using the same start and target points, and the maximum particle count to be 20 particles, the PSO algorithm is tested, and the results are shown in Table 4.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

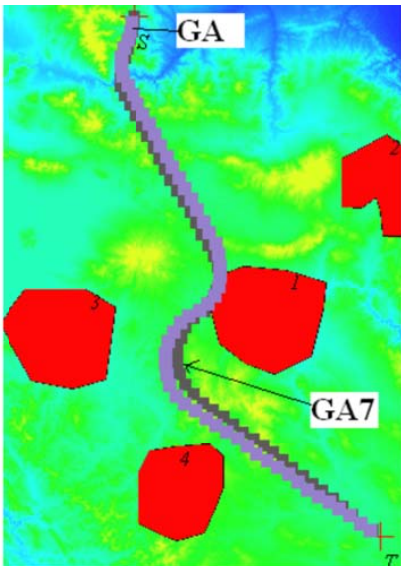


Figure 7. GA trajectories.

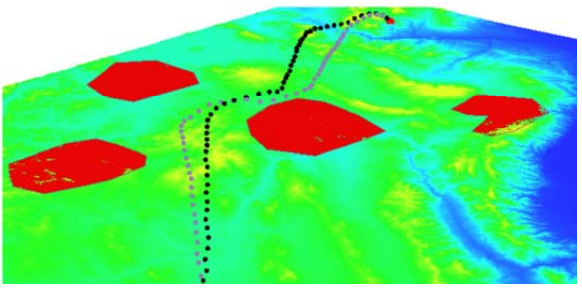


Figure 8. 3D representation of GA trajectories.

Table 4
The results of the execution of the PSO algorithm

Name	Points count	Length	Calc time (m)	Time to reach target (Sec)	Cost (Sec)
PSO	63	503,280	56	1,860	217,132,086·5
PSO 1	57	475,710	51·5	1,680	124,785,824·1
PSO 2	58	490,140	52·5	1,710	110,249,594
PSO 3	59	492,420	53·5	1,740	120,488,710
PSO 4	57	480,330	51·5	1,680	91,051,760
PSO 5	57	478,530	51·3	1,680	63,021,741·1
PSO 6	59	489,120	53·1	1,740	121,135,090·7
PSO 7	60	490,530	53·4	1,770	156,138,674
PSO 8	57	468,660	51	1,680	123,907,850·7
PSO 9	57	471,660	51·5	1,680	134,386,337·2

The best trajectory is PSO5 which has the least cost 63,021,741·1. This trajectory is represented in Fig. 9. Figure 10 shows a 3D representation of the best PSO trajectory:

Each algorithm has generated a near-optimal trajectory. Table 5 compares the two best trajectories generated by GA and PSO algorithms.

GA algorithm generated the better trajectory during the longest period of calculation time but PSO generated a trajectory which is not very different from the trajectory generated by GA using an acceptable period of calculation time (51·3 seconds).

These results and values are changeable according to the area and constraints, but in general the PSO algorithm is faster than GA, but the trajectory generated by the genetic algorithm is better.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

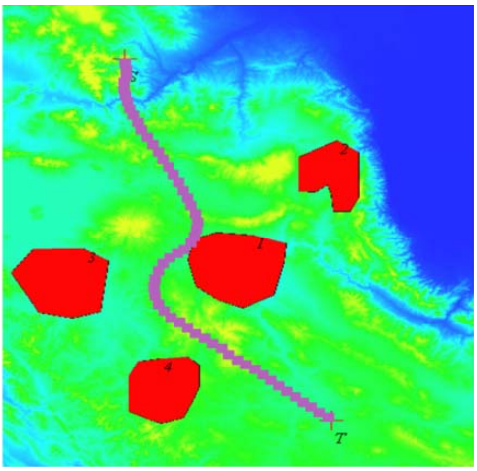


Figure 9. PSO trajectories.

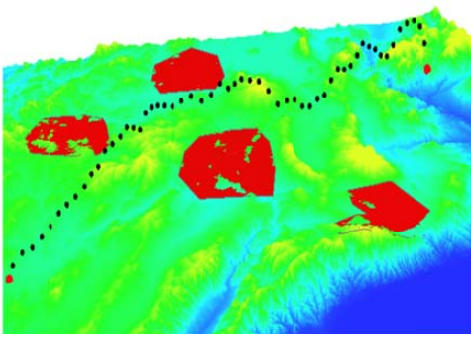


Figure 10. 3D representation of PSO trajectories.

Table 5
Comparing results

Name	Points count	Length	Calc time (m)	Time to reach target (Sec)	Cost (Sec)
GA	59	487,950	63.8	1,740	928,141·5
PSO 5	57	478,530	51.3	1,680	63,021,741·1

8.0 CONCLUSION

This paper presented two evolutionary approaches to solve the trajectory planning problem: GA and PSO algorithms. The optimal trajectory was planned considering three dimensional terrain environment with static and moving obstacles which must be avoided. Simulation results show that each approach finds a near-optimal solution in a very effective and easy manner. GA does more calculations than PSO, but the solution obtained by GA is better, while PSO algorithm needs less calculations, the solution it finds is worse. For further research in this area, it is suggested to combine genetic and fuzzy to make genetic-fuzzy algorithm, or PSO and fuzzy (PSO-fuzzy) to find more improved algorithms, this may be done by using GA, or PSO to construct fuzzy rules, and then build a fuzzy system that uses these rules to solve the trajectory planning problem, this way GA or PSO will generate the best route (the least cost route) using a very fast fuzzy system, this is called genetic-fuzzy algorithm or PSO-fuzzy.

ACKNOWLEDGEMENTS

This work is supported by a grant from University of Guilan, Rasht, Iran. The authors appreciate the valuable comments provided by the anonymous referees.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press

REFERENCES

1. Chen, X.-G. Gao, X.-W. and Qing, D. A genetic-algorithm-based approach to UAV path planning problem, WSEAS Int Conference on Simulation, Modeling and Optimization, 17-19 August 2005, Corfu, Greece, pp 503-507.
2. Bellman, R. *Dynamic Programming*, 1962, Princeton University Press, Princeton, NJ, USA.
3. Betts, J.T. and Huffman, W.P. Path constrained trajectory optimization using sparse sequential quadratic programming, *J Guidance, Control and Dynamics*, January-February 1993, **16**, (1), pp 59-68.
4. Rippel, E., Aharon, B.-G. and Nahum, S. Fast graph-search algorithms for general aviation flight trajectory generation, Technion, 24 May 2004, Israel Institute of Technology, Haifa, Israel.
5. Betts, J.T. and Huffman, W.P. Path constrained trajectory optimization using sparse sequential quadratic programming. *J Guidance, Control and Dynamics*, January-February 1993, **16**, (1), pp 59-68.
6. Enright, P.J. and Conway, B.A. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming, *J Guidance, Control and Dynamics*, July-August 1993, **15**, (4), pp 994-1002.
7. Hall, R. Path planning and autonomous navigation for use in computer generated forces, 2007, Scientific Report, Swedish Defense Research Agency.
8. Lu, P. Inverse dynamics approach to trajectory optimization for an aerospace plane, *J Guidance Control and Dynamics*, July-August 1993, **16**, (4), pp 726-732.
9. Hwang, Y. and Ahuja, N. Gross motion planning – a survey, *ACM Computing Surveys*, September 1992, **24**, (3), pp 219-291.
10. Latombe, J. *Robot Motion Planning*, 1991, Kluwer, Boston, MA, USA.
11. Haeberling, C. Symbolization in topographic 3D-maps: conceptual aspects for user-oriented design, 1999, Zurich Institute of Cartography, Swiss Federal Institute of Technology (ETH Zurich).
12. Zardashti, R. and Bagherian, M. A new model for optimal TF/TA flight path design problem, *Aeronaut J*, May 2009, **113**, (1143), pp 301-308.
13. Anargyros, N.K., Ioannis, K.N., Tsourveloudis, N.C. and Valavanis, K.P. Evolutionary algorithm based on-line path planner for UAV navigation, 10th Mediterranean Conference on Control and Automation, 9-12 July, 2002, Lisbon, Portugal.
14. Ioannis, K.N. and Kimon, P.V. Evolutionary algorithm based offline/online path planner for UAV navigation, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, December 2003, **33**, (6).
15. Call, B.R. Obstacle avoidance for unmanned air vehicles, December 2006, Brigham Young University, USA.
16. Qing, L., Wei, G., Yuping, L. and Chunlin, S. Aircraft toute optimization using genetic algorithms, IEEE, USA, 2-4 September 1997, pp 394-397.
17. Back, T. *Evolutionary Algorithms in Theory and Practice*, 1996, Oxford University, UK
18. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1998, Addison Wesley, Reading, MA, USA.
19. Mitchell, M. *An Introduction to Genetic Algorithms*, 1999, Massachusetts Institute of Technology, Cambridge, MA, USA.
20. Pellazar, M.B. *Vehicle Route Planning With Constraints Using Genetic Algorithms*, 1994, IEEE, Washington, USA.
21. Eberhart, R. and Kennedy, J. A New optimizer using particle swarm theory, 1995, Sixth International Symposium on Micro Machine and Human Science, IEEE, Washington, DC, USA.
22. Coelho, L.dos. S. and Sierakowski, C.A. A software tool for teaching of particle swarm optimization fundamentals, Science Direct, 26 March 2008, Parana, Brazil.
23. Lee, C.-Y. and Shen, Y.-X. Optimal planning of ground grid based on particle swam algorithm, 2009, World Academy of Science, Engineering and Technology, China.
24. Yuhui, S. Particle swarm optimization, February 2004, IEEE Neural Networks Society, USA.
25. Lazinica, A. *Particle Swarm Optimization*, 2009, InTech, Vienna, Austria.
26. Roberge, V., Tarbouchi, M. and Labonte, G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, industrial informatics, *IEEE Transactions*, 2012 **9**, (1), pp 132-141.
27. Wang, Q., Zhang, A. and Qi, L. Three dimentional path planning for UAV based on improved PSO algorithm, 2014, IEEE 26th Chinese Control and Decision Conference, pp 3981-3985.

https://doi.org/10.1017/S0001924000011246 Published online by Cambridge University Press