Constrained optimal terrain following/threat avoidance trajectory planning using network flow

R. Zardashti
rezazardashti@dena.kntu.ac.ir
Faculty of Aerospace Engineering,
K.N. Toosi University of Technology
Tehran, Iran

A. A. Nikkhah
nikkhah@kntu.ac.ir
Faculty of Aerospace Engineering
K.N. Toosi University of Technology
Tehran, Iran

M. J. Yazdanpanah
yazdan@ut.ac.ir
Control and Intelligent Processing Center of Excellence (CIPCE)
School of Electrical and Computer Engineering
College of Engineering, University of Tehran
Tehran, Iran

ABSTRACT

This paper focuses on the trajectory planning for a UAV on a low altitude terrain following/threat avoidance (TF/TA) mission. Using a grid-based approximated discretisation scheme, the continuous constrained optimisation problem into a search problem is transformed over a finite network. A variant of the Minimum Cost Network Flow (MCNF) to this problem is then applied. Based on using the Digital Terrain Elevation Data (DTED) and discrete dynamic equations of motion, the four-dimensional (4D) trajectory (three spatial and one time dimensions) from a starting point to an end point is obtained by minimising a cost function subject to dynamic and mission constraints of the UAV. For each arc in the grid, a cost function is considered as the combination of the arc length, fuel consumption and flight time. The proposed algorithm which considers dynamic and altitude constraints of the UAV explicitly is then used to obtain the feasible trajectory. The resultant trajectory can increase the survivability of the UAV using the threat region avoidance and the terrain masking effect. After obtaining the feasible trajectory, an improved algorithm is proposed to smooth the trajectory. The numeric results are presented to verify the capability of the proposed approach to generate admissible trajectory in minimum possible time in comparison to the previous works.

Paper No. 4002. Manuscript received 13 May 2013, revised version received 4 October 2013, accepted 23 January 2014.

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

NOMENCLATURE

(i,j)	a network node
$((i,j),(k,j\text{-}1))$	a network arc
$C_{k,j\text{-}1,i,j}$	the cost associated with arc $((i,j),(k,j\text{-}1))$
$p(i,j)$	the label of the node (i,j)
$X_{k,j\text{-}1,i,j}$	amount of the flow on arc $((i,j),(k,j\text{-}1))$
α	angle-of-attack, deg
D	drag force, N
g	gravity acceleration, N
γ	flight path angle, deg
L	lift force, N
m	mass of aircraft, kg
μ	bank angle, deg
n_z	load factor
q	dynamic pressure, N/m^2
R	turn radius, m
ρ	air density, kg/m^3
ROC	Rate of Climb, ms^{-1}
S	reference area, m^2
SFC	Specific Fuel Consumption, kg/hour
T	thrust force, N
V	aircraft velocity, ms^{-1}
χ	turn (Heading) angle, deg
$\dot{\chi}$	turn rate, deg/s
ACO	Ant Colony Optimisation
AS	Admissible Set
CRG	Candidate Route Generator
DTED	Digital Terrain Elevation Data
GA	Genetic Algorithm
MCNF	Minimum Cost Network Flow
MSA	Modified Smoothing Algorithm
PSO	Particle Swarm Optimisation
UAV	Unmanned Aerial Vehicle

1.0 INTRODUCTION

A central task in aircraft mission planning is selecting the best route for the mission. Route planning is an important problem for a number of diverse applications including intelligent transportation systems, space applications, autonomous robotics and aircraft guidance and navigation systems[1-9]. Past approaches to automated route planning have usually formulated the problem as a matter of numeric cost minimisation to be solved by methods like dynamic programming[10], genetic algorithm[11], numerical procedures[6,7,12-15] or geometric algorithms like A* search[16,17]. Most geometrical route planning algorithms can be considered either graph-based or grid-based, each with their own associated advantages and disadvantages. Graph-based approaches are usually very accurate, but since some missions require several constraints on the resultant path, such as not exceeding maximum two bank angle and threshold comfort level and/or workload of the autopilot,

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

if these constraints are involved, they face with some difficulties such as long convergence time[18]. Grid-based techniques use a digitised grid consisting of square cells of equal size, representing the environment in which the route planning is performed. The environment is discretised in $m \times n$ grid cells. The best route is chosen as the one from start to goal which has the lowest total cost measured in terms of fuel usage and survivability. A*-based algorithms are variants of the shortest path algorithm. Grid-based approaches converge in real-time but have difficulties with route constraints. Most operational applications use DTED† as input to their route planner. A cost estimation is assigned to a particular cell corresponding to the cost incurred by traveling through that particular region. Then a route is planned from a given start grid cell location to a goal grid cell location with the least total cost. CRG is based on a best-path first, tree search which uses an algorithm to find the least-cost path from a given initial point on a graph to a goal point on a graph. The geographic map area of interest over which the UAV is to fly is overlaid with a graph with a certain pre-defined density of squares. The intent of the graph is to assign a cost to every possible movement between two cells in the graph. This score must reflect a complex set of objectives for the overall route. Therefore, the computation of this set of costs makes up a significant portion of the CRG algorithm. Given the graph, the associated costs, and the start and end points for a UAV route, the CRG algorithm analyses candidate routes that are made up of smaller segments that cross edges of the graph squares or diagonals of the graph squares. This analysis is performed according to a segment cost minimisation function. CRG attempts to algorithmically obtain the lowest total route cost. The route that best minimises overall cost in the graph space is assumed to be the optimal route for the real-world UAV[9]. This approach is 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 search techniques suffer from their relatively slow execution. Graph search methods are a standard tool for shortest-path problems. However, existing work does not address problems with dynamic aspects, namely when accelerations are taken into account as part of the problem constraints and cost[20]. Graph search methods are closely related to Dynamic Programming, but employ the special structure of the shortest-path problem to enhance computational efficiency. Some researchers have formulated the optimal path planning problem as an optimal control problem which is equivalent to a two-point boundary value problem[6,7]. As discussed in[12] this approach is problematic, since the adjoin equations are frequently nonlinear and obtaining them is unwieldy 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 equation; 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[13,21,22]. However, their approaches suffer from long convergence time. The interested reader is referred to[3,4] for more discussion about different route planning approaches.

In work[23], a model of the route planning problem based on a layered network flow structure and shortest path criterion, was presented. By using the kinematic model of a UAV, only kinematic constraints of the UAV were considered. Naturally, this scheme may not provide a feasible flight trajectory, because time-varying dynamic constraints of the UAV are not considered.

In this work, an improved model of the layered network flow structure is introduced which aims to construct a multi-objective optimal and constrained 4D flight trajectory. A 4D motion plan

† A term used by the US Defense Mapping Agency (DMA). It emphasises the measurement of height above a datum and the absolute altitude or elevation of the points over the grid-based data terrain[19].

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

comprises of a georeferenced sequence of 3D waypoints and the desired track velocities between them. In this paper, such tracks are also equivalently referred to as trajectory segments. Based on using the DTED and discrete dynamic equations of motion, the 4D trajectory from a starting point to an end point is obtained by minimising a cost function subject to dynamic and mission constraints of the UAV. For each arc in the grid, a cost function is considered as the combination of the arc length, fuel consumption and flight time. A constrained MCNF algorithm which considers the dynamics and mission constraints of the UAV explicitly is then used to obtain the feasible route. Furthermore, the resultant trajectory can increase the survivability of the UAV by considering the known threat regions and utilising the masking of terrain to minimise vulnerability of the UAV. The goal of this work is that obtains a time-depended optimal and deterministic solution, and will be able to take nonlinear cost function in an offline manner with lower computational time. Moreover, unlike other methods in which the constraints are implicit in the cost function, in this approach, constraints are evaluated directly over the network grids. Thus, this has reduced the complexity of the problem. In the numerical results section, a comparison was made between Optimal Control approach and proposed algorithm.

The rest of the paper is organised as follows. In Section 2, the flight mechanics formulation of the problem with cost functional and constraints are presented. In Section 3, the mathematical model and its network flow structure are constructed utilising a constrained MCNF algorithm based on the optimality conditions are proposed. Additional systematic constraints are imposed to the route in Section 4. An improved trajectory smoothing algorithm (relative to the previous work) is proposed to reduce unnecessary nodes and extra altitude changes of the path to reduce the workload of the autopilot in Section 5, and numerical simulations as well as a comparison of proposed algorithm with an optimal control procedure are presented in Section 6. Section 7 ends the paper with conclusions and future work.

2.0 FLIGHT MECHANICS FORMULATION

2.1 UAV model

A nonlinear dynamic model of a UAV flying in a 3D airspace is presented. The UAV is considered as a point in 3D space, and its translational and angular states at time t are defined by the state vector $X(t) = \{V(t),\gamma(t),\chi(t),m(t),x(t),y(t),z(t)\}^T \in G:\Re^7 \rightarrow \Re^7$. The co-ordinates (x,y,z) are taken relative to an inertial reference frame[24]. The state equations may now be written as:

$$m\dot{V} = T\,\mathrm{Cos}\alpha - D - m\,g\,\mathrm{Sin}\gamma \quad \ldots (1)$$

$$m\,V\,\dot{\gamma} = (T\,\mathrm{Sin}\alpha + L)\,\mathrm{Cos}\mu - m\,g\,\mathrm{Cos}\gamma \quad \ldots (2)$$

$$m.V\,\mathrm{Cos}\gamma\,\dot{\chi} = (T\,\mathrm{Sin}\alpha + L)\,\mathrm{Sin}\mu \quad \ldots (3)$$

$$\dot{m} = -\frac{T.SFC}{3{,}600.g} \quad \ldots (4)$$

$$\dot{x} = V\mathrm{Cos}(\gamma)\mathrm{Cos}(\chi) \quad \ldots (5)$$

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

$$\dot{y} = V\,\mathrm{Cos}(\gamma)\mathrm{Cos}(\chi) \quad \ldots (6)$$

$$\dot{z} = V\,\mathrm{Sin}(\gamma) \quad \ldots (7)$$

Where $T = T(V,h)$ and $SFC = SFC(V,h)$ are given as input tabulated data, $D = q.S.CD$ and $L = q.S.CL$, $q = 1/2\rho V^2$ is the dynamic pressure. The lift and drag coefficients CL and CD are functions of the angle-of-attack α and are given as input tabulated data.
With the combination of (2) and (3), the bank angle is equal to:

$$\mathrm{Tan}\mu = \frac{V.\mathrm{Cos}\gamma.\dot{\chi}}{V.\dot{\gamma} + g.\mathrm{Cos}\gamma} = \frac{\dot{\chi}}{\frac{\dot{\gamma}}{\mathrm{Cos}\gamma} + \frac{g}{V}} \quad \ldots (8)$$

By definition, the load factor is obtained as follows:

$$n_z = \frac{L + T.\mathrm{Sin}\alpha}{m.g} \quad \ldots (9)$$

Equations (2) and (3) can be rewritten as:

$$\dot{\chi} = \frac{g}{V}\,\frac{n.\mathrm{Sin}\mu}{\mathrm{Cos}\gamma} \quad \dot{\gamma} = \frac{g}{V}\left(n\,\mathrm{Cos}\mu - \mathrm{Cos}\gamma\right) \quad \therefore n_z = \frac{\dot{\gamma}V/g + \mathrm{Cos}\gamma}{\mathrm{Cos}\mu} \quad \ldots (10)$$

The lift force Coefficient can be obtained using Equation (2):

$$L = \frac{mV\,\dot{\gamma} + m\,g\,\mathrm{Cos}\gamma}{\mathrm{Cos}\mu} - T\,\mathrm{Sin}\alpha \quad \ldots (11)$$

Using input aerodynamic data relationship between α and CL ($CL = f(\alpha)$) and combination with Equation (11), the trimmed angle-of-attack corresponding to required lift coefficient can be obtained implicitly at any time.

2.2 Flight model constraints

UAV performance is subject to physical constraints that affect its maneuverability. Additional restrictions may be applied to ensure autopilot comfort. To determine the flight model constraints, both horizontal and vertical channel parameters are considered. Rate of climb, turn rate and load factor are used to limit flight performance parameters as follows:

$$\dot{\chi}_{\max} = 57 \cdot 3 \times \frac{V}{R_{\min}} \quad \ldots (12)$$

ROC_{max} and χ_{max} represent the maximum Rate-of-Climb and maximum turn rate respectively, and $R_{min} = \frac{V^2}{g\sqrt{n_{zmax}{}^2 - 1}}$ is the minimum turn radius of the UAV in which g and n_{zmax} are the gravity and the maximum continuous load factor of the UAV, respectively.

Furthermore, bank angle is a significant constraint in UAVs in which they are flying with Bank-To-Turn control approach. This is because of lateral-directional maneuver limitations and radar altimeter field of view bounds.

Also, there are some constraints on velocity and motor thrust because of their physical limitations.

Finally, the overall dynamic constraints are expressed on the next page:

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

$$|ROC| \le ROC_{max}$$

$$|\dot{\chi}| \le \dot{\chi}_{max}$$

$$|\mu| \le \mu_{max}$$

$$n_{z\,min} \le n_z \le n_{z\,max} \quad \ldots (13)$$

$$T_{min} \le T \le T_{max}$$

$$V_{min} \le V \le V_{max}$$

$$CL_{min} \le CL \le CL_{max}$$

2.3 The cost functional

In the previous work, an optimal path based on a single objective optimisation (shortest path) was obtained. In this work, it will be used a combination of some criteria, for instance, length of the arc, fuel consumption and flight time with the multi-objective optimisation (MOO) approach using an improved MCNF algorithm.

The general MOO problem is defined as follows[25]:

$$\begin{aligned} &\underset{k}{Minimize} : C(x)=[C_1(x), C_2(x), ..., C_v(x)]^T, v = 1,...,3 \\ &s.t : g_l(x) \le 0, \quad l=1,2,...,p, \\ &\qquad h_t(x)=0, \quad t=1,2,...,r, \end{aligned} \quad \ldots (14)$$

Where v is the number of objective functions, p is the number of inequality constraints, and r is the number of equality constraints. $x \in R^n$ is a vector of design variables, where n is the number of independent variable x, $C_v(x) \in R^v$ is a vector of objective functions $C_m(x): R^n \rightarrow R^1$, $m = 1,\ldots,v$. $g_l(x)$ and $h_t(x)$ are inequality and equality constraints, respectively.

The predominant concept in defining an optimal point is that of Pareto optimality (Pareto 1906), which is defined as follows:

Definition: Pareto Optimality: A point, $x^* \in X$, is pareto optimal *iff* there does not exist another point, $x \in X$, such that $C(x) \le C(x^*)$, and $C_m(x) < C_m(x^*)$, $m = 1,\ldots,v$, for at least one function.

Variants of multi-objective optimisation methodologies may be distinguished on the basis of their approach for obtaining the Pareto solution set. The following classification was introduced.

A priori methods: Methods of this category work by reformulating the problem to make it fit within the single criteria framework. The preference relation is therefore defined a priori, without having knowledge on the possible trade-offs between the different objectives. The most common a priori method is probably the weighted sum approach, which works by combining the different costs into a unique scalar function. Another common technique is the constraint method, which prescribes to keep a single objective to optimise while moving the others to the constraint side. As a last example, the lexicographic ordering method defines a preference

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

order on the objectives and optimises each of them successively following the order. All these methods have a common characteristic: they define a strict preference order on solutions before the search, and then apply single objective optimisation.

A posteriori methods: A posteriori approach is decomposed into two phases. In the first phase, a specialised algorithm is used to search for a representative set of trade-offs between objectives (generally small). In the second phase, the decision maker picks a final solution from that set using higher order preferences which were not expressed explicitly in the objective functions.

Heuristic methods are also used for multi-objective optimisation; in Ref. 26 applied simulated annealing to multi-objective optimisation, and multi-objective optimisation by genetic algorithms can be found in Refs 27 and 28 among others. It's believed that, heuristic approaches produce near optimal solution because they don't have optimality proof theoretically.

An alternative approach to multi-objective path planning is to use a multi-objective search algorithm like Refs 29 and 30. However, these algorithms are computationally expensive and, in the case of Ref 30, restricted to acyclic graphs; note that graphs derived from grids are cyclic.

The most common approach for the solution of MOO is to transform this into a single objective optimisation problem. This is termed, as mentioned above, the weighted sum method. Thus, a simple total cost functional is used as below:

$$C(t) = \sum_{1}^{v} W_i C_i = W_t \int dt + W_m \int \dot{m} dt + W_s \int \dot{s} dt \qquad \ldots (15)$$

Where W_t,W_m and W_s represent weights of time, fuel and length criterion, respectively; and typically are set by the decision maker such that $\sum W_i = 1$ and $W_i > 0$, $i = 1,\ldots,v$. In this way, by definition, the minimum of Equation (15) is Pareto optimal[25]; i.e., minimising Equation (15) is sufficient for Pareto optimality.

Moreover, if different objective functions have different units, the norm of any degree becomes insufficient to represent closeness mathematically. Consequently, the objective functions should be transformed such that they are dimensionless. Therefore, a common function transformation is presented as follows[25]:

$$C_i^{trans} = \frac{C_i(x)}{\left|C_i^{\max}\right|} \qquad \ldots (16)$$

which results in a non-dimensional objective function with an upper limit of one (or negative one) and an unbounded lower limit (note that $C_i^{max} \neq 0$ is assumed)). Detailed computation of C_i^{max} is discussed in the next section. A comprehensive evaluation of the decision objectives and decision variables for the UAV flight planning is provided in Ref. 31.

In optimal trajectories, minimising the cost functional (Equation (15)) is interesting. Numeric solutions of continuous dynamic programming methods may require multiple passes over the entire state space. For the specific case of the network flow problem, more efficient alternatives are offered by appropriate discrete schemes coupled with an efficient network search procedure.

3.0 ALGORITHM DEFINITIONS AND MODELING

In particular, the model over a special region is discretised, which includes spatial position and the trajectory optimisation problem is formulated as a constrained MCNF problem over a layered

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

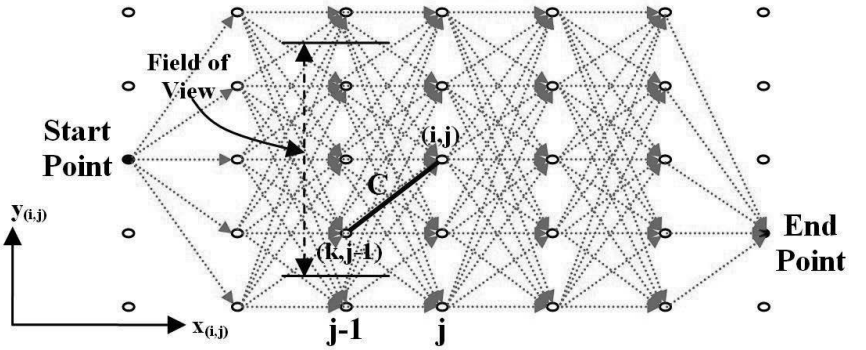


Figure 1. The layered network constructed on DTED.

network flow structure. In this work, a digitised grid consisting of rectangle cells is used to represent the route planning environment. A DTED which consists of an $m \times n$ grid is used as input to the algorithm. A network flow structure is constructed based on the grid of DTED. Each of the grid points is referred to as a node of the network. Since the DTED grid is regular, the nodes can be classified as a layered network with n layers and each layer contains m nodes. Naturally, there are $m \times n$ nodes. Since finding the optimised path is desired, UAV should fly always toward the goal node. Thus each node in the j^{th} layer is connected to the previous nodes in the $(j\text{-}1)^{th}$ layer via a K arcs that K is represents number of nodes in a predefined field of view, so arc $((k,j\text{-}1),(i,j))$, $k \in K$ connects the i^{th} node in the j^{th} layer ,as the tail of the arc, to the k^{th} node in the $(j\text{-}1)$th layer, as the head of the arc (Fig. 1). It would be noted that each node (i,j) represents the position co-ordinate of the form $(x_{(i,j)},y_{(i,j)},z_{(i,j)})$ with respect to inertial reference frame. In essence, the algorithm proceeds from the start (or initial) node and iteratively develops the front of equi-distant nodes. Thus, at each step nodes of one layer are tagged with their optimal cost label, until the goal node is reached. As will be seen, the sequential structure of this scheme will allow us to introduce some important modifications to the algorithm.

Consider a directed graph $G = (V,A)$, where V is a set of $m \times n$ nodes and A represents a set of $V \times V$ arcs. Note that all of the arcs connect nodes of two consecutive layers. The minimum cost network flow (MCNF) problem is about finding a feasible flow such that the sum of the transmission costs is being minimum and can be formulated as follows[20]:

$$\begin{aligned} &Minimize\ J = \sum_{\{(i,j)\in N\}} \sum_{\{k\in K\}} C_{k,j\text{-}1,i,j} X_{k,j\text{-}1,i,j} \\ &s.t. \begin{cases} \sum_{\{[(k,j\text{-}1),(i,j)]\in A\}} X_{k,j\text{-}1,i,j} - \sum_{\{[(k,j\text{-}1),(i,j)]\in A\}} X_{i,j,k,j\text{-}1} = b_{i,j} \\ 0 \le X_{k,j\text{-}1,i,j} \le U_{k,j\text{-}1,i,j} \quad , \quad \{[(k,j\text{-}1),(i,j)] \in A\} \end{cases} \end{aligned} \qquad \ldots (17)$$

Where $X_{k,j\text{-}1,i,j}$ denotes the flow on arc $((k,j\text{-}1),(i,j))$ and $C_{k,j\text{-}1,i,j}$ is the cost to transport one unit of flow on that arc. For each arc, $b_{i,j}$ denotes the flow produced or consumed at node (i,j). If $b_{i,j} > 0$, node (i,j) is called a source (supply) node. If $b_{i,j} < 0$, node (i,j) is called a sink (demand) node. Otherwise (i.e. $b_{i,j} = 0$), node (i,j) is called a transshipment node. For each arc $((i,j),(k,j\text{-}1)) \in A$, $U_{k,j\text{-}1,i,j}$) denotes the upper (lower) bound on the flow on arc. After calculating the optimum path, if arc $((k,j\text{-}1),(i,j))$ is on the resultant path, then $X_{k,j\text{-}1i,j} > 0$, otherwise $X_{k,j\text{-}1,i,j} = 0$.

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

3.1 MCNF algorithm

To solve the optimisation problem Equation (17), the flow $X_{k,j\text{-}1,i,j}$ for all $((k,j\text{-}1),(i,j))\in A$ must be found to minimise the cost function. A variant of the MCNF algorithm is proposed, which is suitable for layered networks. This algorithm uses the optimality conditions of a modified shortest path problem to find the optimal route. To express the optimality conditions, a cost label is associated with each node of the network, which represents the amount of cost from the start node to that node. The cost label of node (i,j) is referred to as $p(i,j)$.

3.1.1 The optimality condition

Theorem (4-1). The necessary and sufficient condition for the optimality of the cost labels is that for all nodes (i,j) and $(k,j\text{-}1)$ which are connected via arc$((k,j\text{-}1),(i,j))$;

$$p(i,j) \le p(k,j-1) + C_{k,j-1,i,j} \quad \ldots (18)$$

Moreover arc $((k,j\text{-}1),(i,j))$ is on the optimal path if and only if;

$$p(i,j) = p(k,j-1) + C_{k,j-1,i,j} \quad \ldots (19)$$

Proof. The interested reader can find the proof of the theorem in Ref. 20.

3.1.2 Computing the optimal cost labels

To start the algorithm, the cost label of the start node is set to zero and the cost label of any other nodes is set to infinity. Then a forward scheme is applied to calculate the cost labels of nodes in each layer considering the optimality conditions and the cost labels of the previous layer. Supposing that the optimal cost labels of the $(j\text{-}1)^{th}$ layer have been computed, for each node k in $(j\text{-}1)^{th}$ layer, the value of any constrained parameter is calculated. If the resulted value does not satisfy parameter's bound, the value of the label (i,j) is set to infinity and the flow on the arc is set to zero, otherwise:

$$p(i,j) = \min_{k}[p(k,j-1) + C_{k,j-1,i,j}] \ , \ X_{k,j-1,i,j} = 1 \quad \ldots (20)$$

According to Equations (18) and (19), $p(i,j)$ is the optimal cost label of node (i,j). Following this scheme, the optimal cost label of all nodes including the goal node can be computed. Note that since the computed cost labels satisfy the optimality conditions in Equatons (18) and (19), they represent the optimum cost $C_{k,j\text{-}1,i,j}$ from the start node to those nodes.

3.1.3 Determining the optimal route

After computing the optimal cost label of the goal node, the optimal route can be determined as follows. Let (l,m) be the goal node of the network and $p(l,m)$ be its optimal cost label. Let $(r,m-1)$ be some node in $(m\text{-}1)^{th}$ layer for which;

$$p(l,m) = p(l,m-1) + C_{r,m-1,l,m} \quad \ldots (21)$$

Then according to Equation (19) arc $((r,m-1),(l,m))$ is on the optimal path. By applying a backward scheme, the arc $((u,m-2),(r,m-1))$ in $(m-2)^{th}$ layer can be found for which $p(r,m-1) = p(u,m-2) + C_{u,m-2,r,m-1}$.

Continuing this backward scheme, it's finally reached to the start node. This form is appropriate for the application of the layered network algorithm, which is an efficient single-pass algorithm. This means that the proposed algorithm can find the optimal path with passing the solution space at once.

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

3.2 Discrete model formulation

In order to implement the layered network using the basic tool of network search, discrete space of the model must be provided. By calculating the arc length (the Euclidean or geometric distance between the two nodes), flight path and heading angles based on nodes geometry may be computed as follow:

$$\begin{aligned}
\Delta s_{(k,j-1,i,j)} &= \sqrt{(x_{(i,j)} - x_{(k,j-1)})^2 + (y_{(i,j)} - y_{(k,j-1)})^2 + (z_{(i,j)} - z_{(k,j-1)})^2} \\
\gamma_{(k,j-1,i,j)} &= \mathrm{Tan}^{-1}\left(\frac{z_{(i,j)} - z_{(k,j-1)}}{\sqrt{(x_{(i,j)} - x_{(k,j-1)})^2 + (y_{(i,j)} - y_{(k,j-1)})^2}} \right) \\
\chi_{(k,j-1,i,j)} &= \mathrm{Tan}^{-1}\left(\frac{y_{(i,j)} - y_{(k,j-1)}}{x_{(i,j)} - x_{(k,j-1)}} \right)
\end{aligned} \quad \ldots (22)$$

By replacing time versus length, one gets:

$$V_{(k,j-1)} = \frac{\Delta s_{(k,j-1,i,j)}}{\Delta t} \Rightarrow \Delta t = \frac{\Delta s_{(k,j-1,i,j)}}{V_{(k,j-1)}} \quad \ldots (23)$$

An expression for rate of climb may be obtained as:

$$ROC_{(k,j-1)} = V_{(k,j-1)}.\mathrm{Sin}\gamma_{(k,j-1,i,j)} \quad \ldots (24)$$

And expressions for rate of flight path angle and heading angle are obtained using finite deference approximation as:

$$\begin{aligned}
\dot{\gamma}_{(k,j-1)} &\cong \frac{\Delta\gamma_{(k,j-1,i,j)}}{\Delta s_{(k,j-1,i,j)}}.V_{(k,j-1)} \\
\dot{\chi}_{(k,j-1)} &\cong \frac{\Delta\chi_{(k,j-1,i,j)}}{\Delta s_{(k,j-1,i,j)}}.V_{(k,j-1)}
\end{aligned} \quad \ldots (25)$$

where $\Delta\gamma_{(k,j-1,i,j)} = \gamma_{(k,j-1,i,j)} - \gamma_{(kk,j-2,k,j-1)}$ and $\Delta\chi_{(k,j-1,i,j)} = \chi_{(k,j-1,i,j)} - \chi_{(kk,j-2,k,j-1)}$, where node ($kk$,$j$-2) is parent (predecessor) of node (k,j-1).
Also, the expression for bank angle is obtained as bellow:

$$\mathrm{Tan}\mu_{(k,j-1)} = \frac{\dot{\chi}_{(k,j-1)}}{\frac{\dot{\gamma}_{(k,j-1)}}{\mathrm{Cos}\gamma_{(k,j-1,i,j)}} + \frac{g}{V_{(k,j-1)}}} \quad \ldots (26)$$

Therefore, the load factor is obtained:

$$n_{z(k,j-1)} = \frac{V_{(k,j-1)}.\dot{\gamma}_{(k,j-1)}}{g.\mathrm{Cos}\mu_{(k,j-1)}} + \frac{\mathrm{Cos}\gamma_{(k,j-1,i,j)}}{\mathrm{Cos}\mu_{(k,j-1)}} \quad \ldots (27)$$

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

Lift force and angle-of-attack is obtained implicitly as:

$$L_{(k,j-1)} = \frac{m_{(k,j-1)}.V_{(k,j-1)}.\dot{\gamma}_{(k,j-1)} + m_{(k,j-1)}.g.\mathrm{Cos}\gamma_{(k,j-1,i,j)}}{\mathrm{Cos}\mu_{(k,j-1)}} - T_{(k,j-1)}.\mathrm{Sin}\alpha_{(k,j-1)}$$
$$\alpha_{(k,j-1)} = f(CL_{(k,j-1)}) \ , \ \ L_{(k,j-1)} = CL_{(k,j-1)}.q_{(k,j-1)}.S \qquad \ldots (28)$$

Note that the discrete non-dimensional cost functional using Equations (15) and (16) may be rewritten as follows:

$$C_{k,j-1,i,j} = W_t \frac{\Delta t_{(k,j-1,i,j)}}{\Delta t_{\max}} + W_m \frac{\Delta m_{(k,j-1,i,j)}}{\Delta m_{\max}} + W_s \frac{\Delta s_{(k,j-1,i,j)}}{\Delta s_{\max}} \qquad \ldots (29)$$

where:

$$\Delta t_{\max} = \max_k \frac{\Delta s_{(k,j-1,i,j)}}{V_{(k,j-1)}} \ , \ \ \Delta s_{\max} = \max_k \Delta s_{(k,j-1,i,j)}$$
$$\Delta m_{\max} = \max_k \frac{T.SFC.\Delta s_{(k,j-1,i,j)}}{3600.g.V_{(k,j-1)}} \ , \qquad \ldots (30)$$
$$T = T(z_{(i,j)}, V_{(k,j-1)}) \ , \ SFC = SFC(z_{(i,j)}, V_{(k,j-1)})$$

Then, the final cost function is obtained:

$$C_{k,j-1,i,j} = \Delta s_{(k,j-1,i,j)} \left\{ \frac{W_t}{\Delta t_{\max}.V_{(k,j-1)}} + \frac{W_m.T.SFC}{V_{(k,j-1)}.\Delta m_{\max}} + \frac{W_s}{\Delta s_{\max}} \right\} \qquad \ldots (31)$$

With parametric dependencies and above expressions, the total cost functional is defined as $C = f(\Delta S, V)$, where f shows the cost functional of Equation (31). In other words, velocity and arc length parameters are used for the optimality search. Moreover, referring to the previous section, there are ten equality constraint and eight inequality constraint to solve the constraint optimal problem.

4.0 IMPOSING ADDITIONAL CONSTRAINTS

In realistic scenarios where the UAV is deployed in low altitude to avoid detectability and to use terrain masking, it can be imposed altitude constraint to the generated path. For this purpose it is sufficient to do as follows. First a threshold value for maximum flyable altitude for the UAV should be specified. Then while computing the optimal cost labels, if the altitude of some node is greater than the threshold value, by setting its optimal cost label to infinity, the designed path doesn't pass through regions with altitudes greater than the threshold value.

Using this technique, it can be also expressed the threat regions as areas with the altitudes greater than the threshold value. This causes the generated path to avoid threat regions automatically. Therefore, one of the common threat models is a circular model which is defined as follows:

$$f_{threat}(x,y) = \begin{cases} (x - x_T)^2 + (y - y_T)^2 = {R_T}^2 \\ z \in [0, \infty) \end{cases} \qquad \ldots (32)$$

where (X_T, Y_T) and R_T represent centre and radius of the circular threat region, respectively. The physical aspect of this model implies that entering to this threat region at any conditions is impossible. Therefore, its internal nodes are considered as a hard constraint.

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

5.0 DESIGNING THE ALTITUDE SMOOTHING ALGORITHM

In order to reduce broken (stair- or wiggle-shaped) flight path effects due to discrete construction, it is necessary to smooth the path somehow to eliminate unwanted altitude changes beside lateral changes. The main concern for this problem is the autopilot workload. During the flight, the autopilot is preoccupied with numerous tasks on top of the need to follow the outlined trajectory. Having a rapidly varying trajectory requires the autopilot to constantly focus on the task of following it, especially for low flight over irregular terrain. We shall therefore rely on the riding qualities cost to regulate the smoothness of the planned trajectory. Moreover, the UAV need not always stay close to the ground. Instead, it is sufficient to stay low enough so as not to be visible to the surrounding threats[††]. This means that there may be regions along the route where the UAV can fly considerably above the terrain following command height and still avoid detection.

To this aim, a dual-phase algorithm has been proposed. Consider the constrained optimal path generated by the proposed algorithm. In the first phase, unnecessary nodes are removed while in the second phase, the algorithm smoothes the altitude changes of the UAV.

5.1 Modified Smoothing Algorithm (MSA)

Consider the constructed constrained optimal path. The candidate path is described by a list of waypoints WPs.

Phase 1- Node reduction:

1. Take a threshold value for admissible turn angle changes.
2. Define an empty *AS* consisting of admissible *WPs* of the path and add the start point and the goal point to *AS*.
3. Take a WP_i on the path with its neighbor waypoints, WP_{i-1} and WP_{i+1}.
4. Determine the ascending and descending of the intervals.
5. Compute the turn angle changes between the intervals in the stage of 4.
6. For the WP_i, if the turn angle difference computed in stage 5 is greater than the threshold value, add the WP_i to *AS*, otherwise, go to stage 3.
7. The desired optimal path passes through the points of *AS*.

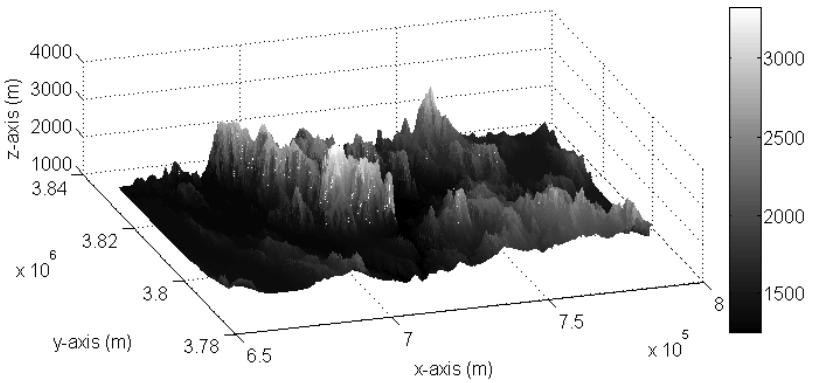


Figure 2. A real DETD map area loaded in MATLAB programming.

[††] Note that a clearance height is defined and added to the *z* direction of the path to avoid collision with terrain.

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

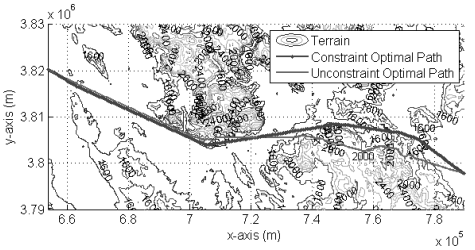


Figure 3. The resultant path without and with dynamic constraints in the same contoured terrain.

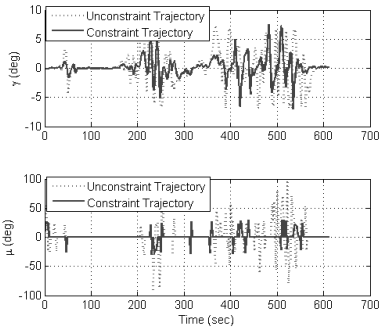


Figure 4. Time history of rate of climb and bank angle of Fig. 3.

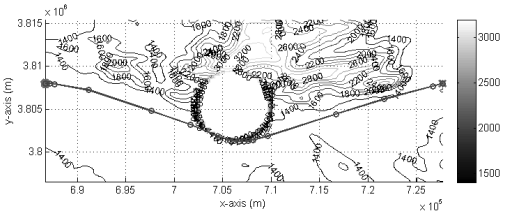


Figure 5. Constrained optimal trajectory with a threat area.

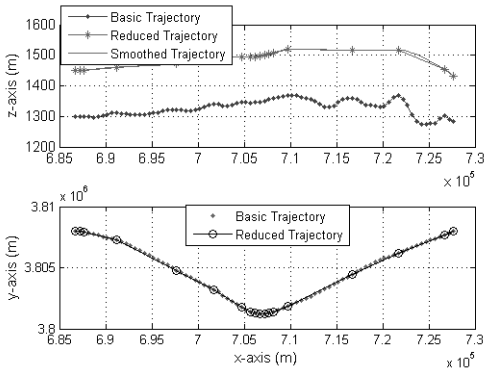


Figure 6. Generated trajectory in *x-z* and *x-y* plane, respectively.

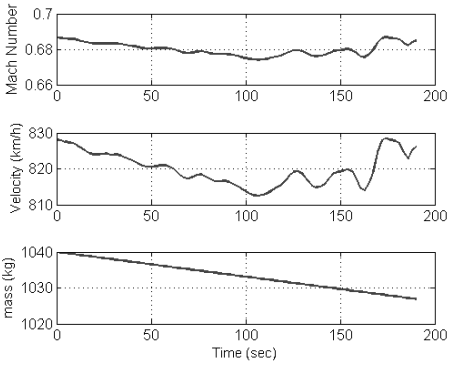


Figure 7. The resultant time history of the Mach no., velocity and mass of the UAV.

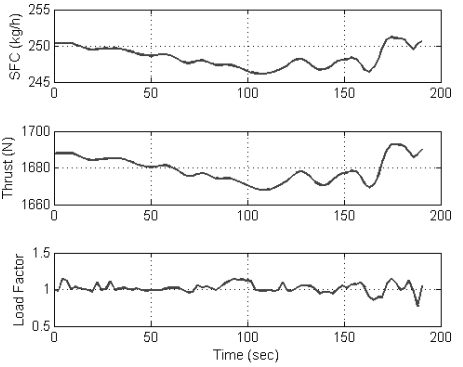


Figure 8. The resultant time history of the fuel consumption, thrust and load factor.

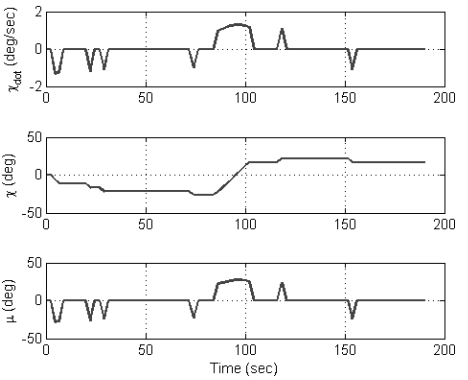


Figure 9. The resultant time history of the turn rate, turn angle and bank angle.

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

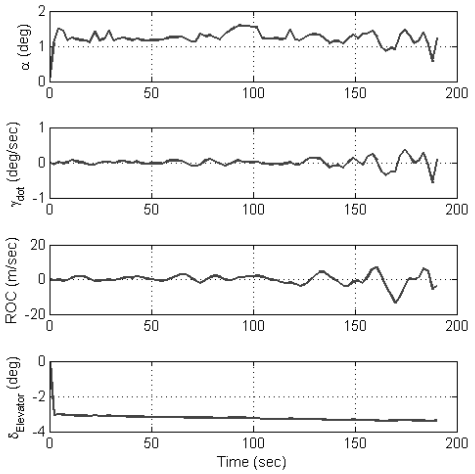


Figure 10. The resultant time history of the angle of attack, fight path angle, rate of climb and elevator angle.

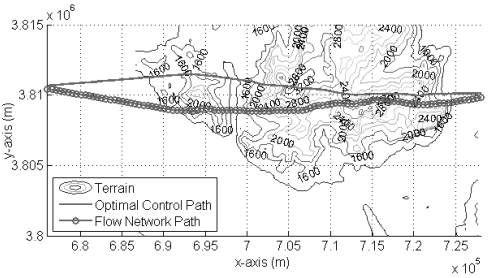


Figure 11. Comparison of trajectories generated by the Optimal Control scheme and the proposed algorithm.

***Phase 2- Altitude smoothing*:**

1. Take a threshold value for admissible altitude changes.
2. Take a WP_i on the AS with its neighbor waypoints, WP_{i-1} and WP_{i+1}, $i \in AS$.
3. Compute the altitude difference between the intervals obtained in stage 2.
4. For that node, if the altitude difference computed in stage 3 is greater than the threshold value of stage 1, update the value of the altitude by averaging altitudes of neighbor points.
5. The final optimal path passes through the points of AS.

The result is a waypoint list AS with fewer and verified waypoints of generated path. The resulted path with less workload of the autopilot is obtained which is more suitable for the UAV.

6.0 NUMERICAL SIMULATIONS

To verify the performance of the optimal trajectory algorithm discussed, numerical simulations are performed. The algorithmic engine is a MATLAB based programming. A real DTED with UTM standard projection (metric co-ordinate system) has been used as input to algorithm. Figure 2 shows the digitised model of this area loaded in MATLAB software. The resolution of this map is 232,050 nodes (m = 850, n = 273). All the experiments were performed on a 3·3-GHz Intel Core-i3 CPU with 4 GB of physical RAM running at 64-bit Microsoft Windows XP.

The algorithm is run with, $\dot{\chi}_{max} = 5°/\text{sec}$, $ROC_{max} = 30\text{ms}^{-1}$, $n_{max} = 3$, $\mu_{max} = 30°$, $h_{max} = 1{,}900\text{m}$ and initial velocity of 230ms^{-1}. Figure 3 shows the comparison of the optimal unconstrained (kinematic constrained) and constrained path generated by the algorithm associated with shortest path criterion ($W_t = 0{\cdot}0$, $W_m = 0{\cdot}0$, $W_s = 1{\cdot}0$) in the map with elevation contours. Figure 4 shows the comparison of the time history of rate of climb and bank angle for unconstrained and constrained paths, respectively. This shows that in the unconstrained path, some dynamic parameters are out

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

of their limitations. The execution time of the algorithm for this resolution is about 85 seconds, based on the hardware which is used.

An illustration of a typical MOO planning in the present of a circular threat region and another DTED area with 31,457 nodes (m = 379, n = 83) is depicted in Table 1. Here are three different cases are investigated. From the results, it can be deduced the effectiveness of the weights, even for a smaller DTED. This means that, for instance, choosing $W_t = 0{\cdot}1$, $W_m = 0{\cdot}8$, $W_s = 0{\cdot}1$ gives less fuel consumption than the other two. With this choice, the resultant optimised trajectory illustrated in Fig. 5. Figure 6 shows the same trajectory in x-z and x-y planes. It illustrates node reduced and altitude smoothed trajectory using MSA with a height clearance of about 150m. Figure 7 shows the resultant time history of Mach Number, velocity and mass of the UAV in the trajectory. Figure 8 shows resultant of SFC, thrust and load factor versus flight time. Figure 9 shows resultant of turn rate, turn angle and bank angle versus flight time. Figure 10 shows resultant of angle-of-attack, fight angle, rate of climb and delta elevator versus flight time. It illustrates trimmed results for the trajectory utilising trimmed aerodynamic coefficients.

Table 1
Comparing of results for three objectives

Options	Flight Time, s	Fuel Consumption(kg)	Distance(m)
$Wt = 0{\cdot}1$, $Wm = 0{\cdot}8$, $Ws = 0{\cdot}1$	191·5	13·0	43,190
$Wt = 0{\cdot}8$, $Wm = 0{\cdot}1$, $Ws = 0{\cdot}1$	189·1	13·7	43,182
$Wt = 0{\cdot}1$, $Wm = 0{\cdot}1$, $Ws = 0{\cdot}8$	189·4	13·8	43,180

A comparison of the proposed algorithm with an optimal control procedure[8] is made. Both algorithms are implemented with the same hardware configuration and boundary conditions: $\dot{\chi}_{max} = 4{\cdot}5$deg/s, $\dot{\gamma}_{max} = 4{\cdot}5$deg/s, constant velocity of $V = 234$ms^{-1} and minimum time of flight criterion. Figure 11 depicts a comparison of trajectories generated by optimal control scheme and the proposed algorithm with the same start and end points. The execution time for the proposed algorithm is approximately 80 seconds which for optimal control scheme is about 420 seconds per iteration. This shows that the proposed algorithm is approximately five times faster (on average) than that of the second one: lower time consuming. Moreover, the time of flight produced by the proposed algorithm is about 245 seconds while the optimal control procedure produced about 250 seconds. This shows that the flight time corresponding to the proposed algorithm is 5 seconds less than that of the second one based on minimum time of flight criterion. It will be noted that the optimal control procedure is sensitive to the initial heading angle; Different initial heading angles lead to paths with different end point miss distance from the goal node. So the best initial heading angle should be determined by repeating the algorithm with different heading angles. The best initial heading angle is obtained as the initial heading angle which has the minimum miss distance from end point.

7.0 CONCLUSIONS

In this paper a modified model was proposed to formulate the multi-objective optimal and dynamic constraint 4D trajectory planning for TF/TA flight. This is a continuation of the previous work[23] in which by using only kinematic parameters and single objective optimisation, 3D path was generated. In the proposed work, according to the optimality conditions of a multi objective problem combining flight time, fuel consumption and the path length, a variant of the constrained

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

MCNF algorithm is proposed to solve the problem regarding the dynamic constraints explicitly using 3D equations of motion of the UAV. Numerical simulations are performed and the results compared to those with optimal control procedure[8] was shown. The results show the effectiveness and robustness of this method in producing optimal trajectory. The advantages of proposed algorithm can be referred as follows. First, the proposed algorithm produces optimal, constraint and deterministic four-dimensional trajectory, while many other methods, such as optimal control procedure or geometric procedures of Refs 18 and 32-34 use the simplified equations or produce flight paths in a much longer time. The second advantage is that the procedure doesn't need to calculate and store the cost map as a data bank in comparison to Ref. 9 which is based on this very time consuming task. The third advantage is that the proposed algorithm is capable of taking any constrained parameter using label setting technique in network flow instead of entering the cost function implicitly and increasing the complexity of the solution. As the last advantage, the proposed procedure is able to find an exact and optimal solution of the problem, while the many other procedures such as heuristic procedures (GA, PSO, ACO, etc.[11,34-36]) only find the near optimal solution; and many of them as well as optimal control procedure[8] have a miss distance from the goal (end) point. A future research topic is the inclusion of Inertial Navigation System Error to the algorithm to avoid collision with terrain. An additional research is the inclusion of the risk of threat regions as an objective to the cost functional and solving the problem in such a way that make a decision to enter the threatened regions.

ACKNOWLEDGEMENTS

The authors would like to thank members of Advanced Control Systems Lab of the University of Tehran for the fruitful discussion on the subject. Moreover, the valuable comments provided by the referees who led to this improved version of the paper are appreciated.

REFERENCES

1. AL-HASAN, S. and G. VACHTSEVANOS, G. Intelligent route planning for fast autonomous vehicles operating in a large natural terrain, *Robotics and Autonomous Systems*, 2002, **40**, pp 1–24.
2. BODENHORN, C., GALKOWSKI, P., STILES, P., SZCZERBA, R. and GLICKSTEIN, I. Personalizing onboard route re-planning for recon, attack, and special operations missions, American Helicopter Society Conference (Avionics and Crew Systems Technical Specialists Conference), 1997.
3. HWANG, Y. and AHUJA, N. Gross motion planning – a survey, *ACM Computing Surveys*, 1992, **24**, (3), pp 219–291.
4. LATOMBE, J. *Robot Motion Planning*, Boston, USA, MA: Kluwer, 1991.
5. SZCZERBA, R., CHEN, D., UHRAN, I. and J.A. A framed-quadtree approach for determining euclidean shortest paths in a 2D environment, *IEEE Transactions on Robotics and Automation*, 1997, **13**, (5), pp 668–681.
6. SHARMA,T., WILLIAMS, P., BILL, C. and EBERHARD, A. Optimal three dimensional aircraft terrain following and collision avoidance, *ANZIAM J*, 2007, pp 695–711.
7. VINCENT, T.L. and GRANTHAM, W.J. *Nonlinear and Optimal Control Systems*, John Wiley and Sons, 1999.
8. TWIGG, C.A. and JOHNSON, S.E. 3D trajectory optimization for terrain following and terrain masking, AIAA, Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado, USA, pp 21–24, 2006.
9. www.virtualacquisitionshowcase.com/docs/2008/scientific1-brief.pdf, 2008.
10. D.B.M. and FONTES, M. Optimal trees for general nonlinear network flow problems: A dynamic programming approach, *Robotica*, 2005, **23**, (5), pp 567–580.
11. DU, X., CHEN, H.-H.and GU, W.-K. Neural network and genetic algorithm based global path planning in a static environment, *J Zhejiang University SCIENCE*, 2004.

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press

12. BETTS, J and H.W.P. Path constrained trajectory optimization using sparse sequential quadratic programming, *J Guidance, Control and Dynamics*, 1993, **16**, (1), pp 59–68.
13. HALL, R. Path planning and autonomous navigation for use in computer generated forces, Scientific Report, Swedish Defense Research Agency, 2007.
14. MALAEK, S. and KOSARI, A.R. A novel minimum time trajectory planning in terrain following flight, IEEE Aerospace and Electrical System Conference, **8**, 2003.
15. SZCZERBA, R. New cell decomposition techniques for planning optimal paths. PhD thesis, University of Notre Dame, Notre Dame, France, 1996.
16. HART, P., NILSSON, P. and RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on System, Science and Cybernetics*, 1968, **4**, (2), pp 100–107.
17. RIPPEL, B.-G.-A., E. and SHIMKIN, N. Fast graph-search algorithms for general aviation flight trajectory generation, TechnionIsrael Institute of Technology, Haifa, Israel, 2004.
18. CORMEN, N., LEISERSON, T., RIVEST, R. and STEIN, C. *Introduction to Algorithms*, McGraw Hill, 2002.
19. EL-SHEIMY, C. VALEO, and HABIB, A. Digital Terrain Modeling. Artec House, 2005.
20. RAVINDRA, T.-L.M., AHUJA, K. and JAMES, B.O. *Network flows, Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, 1993.
21. ENRIGHT, P. J. and CONWAY, B.A. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming, *J Guidance, Control and Dynamics*, 1993, **14**, (4), pp 994–2002.
22. LU, P. Inverse dynamics approach to trajectory optimization for an aerospace plane, *J Guidance, Control and Dynamics*, 1993, **16**, (4), pp 726–732.
23. ZARDASHTI, R. and BAGHERIAN, M. A new model for optimal *tf/ta* flight path design problem, *Aeronaut J*, 2009, **113**, (1143), pp 301–308.
24. MIELE, A. *Flight Mechanics, I, Theory of Flight Paths*, Addison-Wesley, Reading MA., USA, 1962.
25. MARLER, R. and ARORA, J. Survey of multi-objective optimization methods for engineering, *Struct Multidisc Optim*, 2004, **26**, pp 369–395.
26. S. P, Simulated annealing for multiobjective optimization problems, in Proc. 10th International Conference on Multiple Criteria Decision Making, **1**, 1992.
27. GOLDBERG, D. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
28. FONSECA, P. and FLEMING, C.M. Genetic algorithms for multiobjective optimization: formulation, discussion, and generalization, The Fifth International Conference on Genetic Algorithms, 1993.
29. FUJIMURA, K. Path planning with multiple objectives, *IEEE Robot Autom Mag*, 1996, **3**, (1), pp 33–38.
30. STEWART, I.C.B.S. and WHITE, C. Multiobjective a*, *J ACM*, 1991, **38**, (4), pp 775–814.
31. WU, D.C.P. and MERZ, T. On-board multi-objective mission planning for unmanned aerial vehicles, in Proc. IEEE Aerosp Conf, Big Sky, MT, USA, 2009, **7**, (14), pp 1–10.
32. HARGRAVES, C.R. and PARIS, S.W. Direct trajectory optimization using nonlinear programming and collocation, *J Guidance, Control and Dynamics*, 1987, **10**, (4), pp 338–342.
33. AKRAM, M., PASHA, A. and IQBAL, N. Optimal path planner for autonomous vehicles, Proceedings of world Academy of Science, *Engineering and Technology*, 2005, **3**, pp 134–137.
34. HELGASON, V.R., KENNINGTON, L.J. LEWIS, and R.K. Cruise missile mission planning: A heuristic algorithm for automatic path generation, *J Heuristics*, 2001, **7**, (1), pp 473–494.
35. ENGELBRECHT, A. *Fundamentals of Computational Swarm Intelligence*, John Wiley and Sons, 2005.
36. DREO, J. and SIARRY, P. An ant colony algorithm aimed at dynamic continuous optimization, *Applied Mathematics and Computation*, 2006, **181**, (1), pp 457– 467.

https://doi.org/10.1017/S0001924000009349 Published online by Cambridge University Press