2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Multi-objective UAV routing

Lucía Hernández-Hernández, Antonios Tsourdos and Hyo-Sang Shin
Centre for Cyber-Physical Systems, School of Engineering, Cranfield University, Cranfield, Bedfordshire, UK

Antony Waldock
BAE Systems Advanced Technology Centre
Bristol, UK

***Abstract*—The purpose of the study, presented in this paper, has been to apply a multi-objective graph-based method for plan routes of a simulated UAV taking into account scenarios with danger zones or prohibited areas, characteristic of civil airspace, as well as areas with time and speed restrictions and some negotiable "safe corridors". In particular, the method that has been used is New Approach to Multi-Objective A* (NAMOA*), which labelled the paths of the scenarios' graphs with cost vectors where each component was a different objective to be minimised, such as distance travelled, angle change, time taken, fuel consumption and deviation from the straight path. The results had shown that NAMOA* properly found a set of non-dominated paths equal to an estimated Pareto-optimal front.**

***Index Terms*— UAV, multi-objective, route planning, optimization, NAMOA*.**

I. Introduction

The Shortest Path (SP) problem is a daily issue that arises both in scopes as complex as Artificial Intelligence (AI) and in ordinary life, when, for example, someone is searching which is the optimal road to arrive by car to a particular place using, for instance, Google Maps© application which provides distance and time information of several roads to go from a starting point to a final one. Most algorithms calculate the optimal solution for a single objective. However, Multi-Objective Shortest Path (MOSP) problems are becoming more important; mainly for its great potential in applications focused on route planning for unmanned vehicles, a currently growing sector. In particular, it is noticeable how the implementation of Unmanned Aerial Vehicles (UAVs) into the civil airspace, is attracting the attention of many researchers and companies; such as BAE Systems, which has pioneering remotely flight Jetstream aircraft in April 2013 through UK airspace within the framework of ASTRAEA programme [1].

This paper begins with a brief introduction to the state of the art of multi-objective route planning, to continue with the definition of the proposed problem in section 3. Sections 4 and 5 present the analysis and results, followed by the problem validation. Finally, section 7, outlines the conclusions and future work.

II. Background

The functionality to automatically plan a route for a UAV given a number of objectives has been demonstrated using Evolutionary Algorithms [2] such as Genetic Algorithms and Particle Swarm Optimisation, but if it is represented as a graph this can also be solved using deterministic algorithms. Such graph-based methods are a common field of study in AI and have studied SP problems. Dijkstra's algorithm [3] provides a solution for that kind of problems by obtaining a minimal cost route between a starting and end point in a network. An extension of Dijkstra's algorithm is the A* [4], a heuristic best-first search algorithm that uses cost estimations to improve efficiency to reach the goal point. In 2006 [5], it was demonstrated the efficient navigation through an irregular environment of an autonomous ground vehicle which used classical A* algorithm to search an easy trajectory to travel with the help of a map. However, this study aims to go beyond including the multi-objective factor, which adds an element of difficulty because of the higher computational complexity.

Realistic problems must consider several simultaneous and conflicting costs. In those MOSP problems, the arcs, that join two points of the network, are labelled with cost vectors, where each component stands for a different objective to be minimised when choosing alternative paths, for example: Time, speed, consumed fuel, goal accuracy, likelihood of detection and distance travelled. These problems in rare occasions have a single optimal solution which minimises all the objectives considered, so, a partial order preference relation called dominance, is induced, i.e. Let f^1 and f^2 be two k-dimensional objective functions cost vectors, f^1 dominates f^2 if and only if $f^1{}_i \leq f^2{}_i$ for $1 \leq i \leq k$ and $f^1 \neq f^2$.

The first multi-objective search problem was studied by Hansen [6] extending Dijkstra's algorithm to a multi-objective context. As well, some multi-objective heuristic extension of A* have been discovered in later

years, such as Multi-Objective A* (MOA*) [7] and New Approach to Multi-Objective A* (NAMOA*) [8]. The latest studies have been focused on the formal analysis of those extensions [9] to highlight the beneficial properties of heuristic information and also have been centred on compare their performance [10]. However, a field that has not been fully developed is the application of these multi-objective graph-based methods to simulated scenarios. Waldock and Corne [11] used NAMOA* to find the optimal path of manned and unmanned ground vehicles through an unstructured environment. None research seems to have been done with NAMOA*, prior to this research, on airspace environments with the restrictions and the obstacles typical of an air traffic network.

III. Problem Formulation

This study aims for applying graph-based methods for multi-objective route planning of an aerial platform, taking into account the carefulness and timeliness required when it flies in a controlled airspace.

From the aforementioned multi-objective heuristic graph search algorithms, NAMOA* has shown some beneficial differences in comparison with MOA* [9], such as that MOA* perform unnecessary path extensions and node re-expansions, increasing the storage requirements for not interesting cost vectors. However, NAMOA* is able to safety avoid that extra operations, without jeopardising other properties. The dominance of NAMOA* over MOA* and the absence of application of NAMOA* into the UAV routing field, have determined NAMOA* algorithm as the heuristic method to be used in this research, and therefore, contribute to fill that sector and show potential uses of the algorithm.

Next sections describe the characteristics of grid maps and define the objective functions considered in the problem.

A. Grid Maps

The graph through which the optimal path has been evaluated is a simulation of a segregated two-dimensional airspace, in which the danger or prohibited areas are previously known, as well as those zones with fixed time and speed constraints. It can be taken into account the possibility of negotiate "safe corridors", in the prohibited zones, through which the aircraft would fly under certain conditions. The area where the vehicle has to move, from a known start point to an also known goal, has been considered available. So, it is provided a previously designed grid map with $|N|$ nodes, given in cartesian coordinates (x,y), connected by $|A|$ flyable paths called arcs (n, n'). That arcs between nodes can be directed or undirected, i.e. (n_i, n_j) does not equal (n_j, n_i). Arcs are labelled with positive costs vectors $\vec{c}\,(n, n') \in \mathbb{R}^k$ where each component got the value of one of the objective functions, in which a path is defined as a sequence of nodes $P = (n_1, n_2, \ldots n_m)$ such that for all $i < m$, $(n_i, n_{i+1}) \in A$, and the cost vector of each path:

$$\vec{c}(P) = \sum_{i=1}^{m-1} \vec{c}(n_i, n_{i+1}) \tag{1}$$

The objective of NAMOA* is to find a set of non-dominated cost paths in the graph from a starting node $n_s \in N$ to a goal node $n_g \in N$ [9].

B. Objective Functions: Distance Travelled, Angle Change and Number of Odd Nodes

- Distance travelled, d (km): This first objective is defined as the summation of the path segments distance, bearing in mind that a single arc distance, from node n_i to n_j is:

$$d_{ij} = \left|\vec{d}_{ij}\right| = \left|(x_j, y_j) - (x_i, y_i)\right| = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{2}$$

- Angle change, α (rad): The angle change is the summation of the angle between the pairs of arcs that constitute a route. To be able to calculate the change in the travelled angle of an arc, it is necessary to know the location of the immediately predecessor arc. For instance, the angle change from arc (n_i, n_j) to arc (n_j, n_k) is α_{ijk} which is calculated knowing the coordinates of those three nodes and takes values between 0 and π rad. It has been assumed that the UAV departs the start node with an undefined angle, that is, the angle change value of the arcs that connects the start node with its successors, is always zero because there is no predecessor node.

$$\alpha_{ijk} = \cos^{-1}\left(\vec{d}_{ij} \cdot \vec{d}_{jk} \Big/ \left(\left|\vec{d}_{ij}\right| \cdot \left|\vec{d}_{jk}\right|\right)\right) \tag{3}$$

- Number of odd nodes: Since all the nodes of the grid are named by a number, this objective function represents the number of nodes, that composed the path, which have been named with odd numbers.

C. Objective Functions: Time Taken, Fuel Consumption and Area Deviation

The UAV chosen for this study has been the Cranfield University Jetstream-31 G-NFLA powered by two turbo-prop engines. However, some assumptions and dynamics simplifications have been made with the purpose of reduce and accelerate the three objective calculations during the evaluation of the map nodes done by NAMOA*:

- The aircraft flies in a two dimensional plane, that is, at constant altitude, which value is not relevant for the study, and also at a constant airspeed V of 92.60 m/s. This speed only changes when the node, to which is flying, is in a speed constraints area. For those cases, the airspeed also remains constant but with a reduced value of 61.73 m/s [12]. It has been proved mathematically, that at the highest speed (92.60 m/s), the Jetstream 31 is capable of performing the turning radios for the more aggressive trajectory, which is the one executed in simulation when the change in the angle between two segments of the route is the maximum established, that is, π rad.
- The fuel consumption is constant, fixing a fuel burn *fb* of 255 kg/h [12].
- The total mass of the whole aircraft m, has been established to 6450 kg, knowing that 7059 kg is the maximum take-off mass [13]. The mass has been assumed constant, without taking into account decreases caused by fuel consumptions.

No take-off and landing calculation have been taken for this study. It has been considered that the UAV runs the algorithm once is located at the altitude of the starting node.

In order to simulate an approximation of the turning movements of the UAV when the path changes the direction, it has been established the derivate of the unit step response of a second order transfer function $G(s)$ as nearly the trajectory described by the UAV when the change in the angle, from one arc to another, is higher or equal to $\pi/8$ rad. If the angle is lower than $\pi/8$ rad, the turn is negligible.

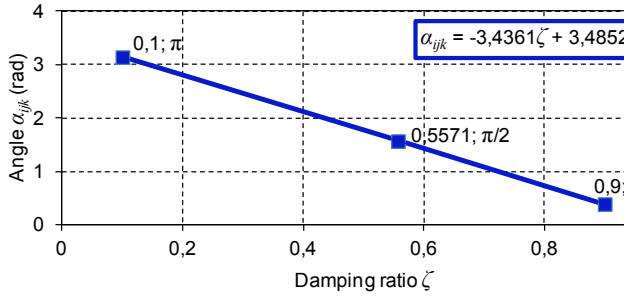

Fig. 1. Dependence of damping ratio (ζ) with angle change (α_{ijk}).

The transfer function $G(s)$, has a constant static gain K_G equal to 1.5 and a constant unit frequency w_n. However the damping ratio ζ directly depends on the angle change α_{ijk} as it has been presented in Fig. 1. So, this dependence causes a change in the derivate of the step response as a function of the angle.

$$G(s) = \frac{K_G w_n^2}{s^2 + 2\zeta w_n s + w_n^2} = \frac{1.5}{s^2 + 2\zeta s + 1} \quad (4)$$

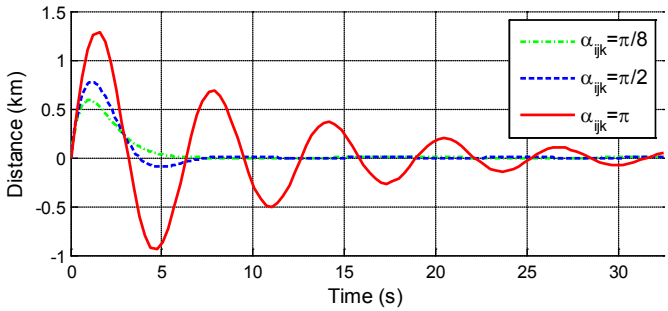

Fig. 2. Comparative of the derivate of the step response for three transfer functions in which the damping ratio is determined by the change in the angle α_{ijk} when moving from one segment of the grid to another.

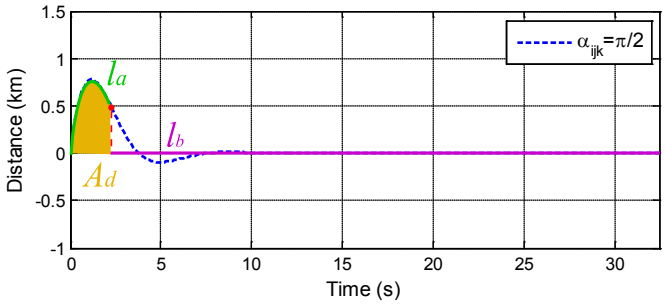

Fig. 3. Representation of the area deviation A_d and the lengths l_a and l_b, of a trajectory segment (n_j, n_k) in which the change in the angle α_{ijk} with respect to the previous segment (n_i, n_j), is $\pi/2$ rad.

The transfer functions are evaluated during a time equal to the minimum time that it takes, to the UAV, to travel the straight line that connects the involved nodes, which is the distance between them divided by the airspeed. For instance, if NAMOA* algorithm is evaluating the costs of moving from node n_j to n_k, being n_i the predecessor of n_j, the value of the angle α_{ijk} will determine the value of the damping ratio, and so the characteristics of the transfer function; and the value of the distance d_{jk} establishes the time equal to d_{jk}/V at which the response is simulated. Figure 2 compares the derivate of the step response of three different transfer functions, which implies three different angle changes α_{ijk}, when the Euclidean distance between the nodes is 3 km, that is, at 92.60 m/s, the step is simulated during 32.43 s.

As a result of the hitherto mentioned, it is possible to describe the following three objective functions:

- Time taken, t (s): This objective is the summation of route distances travelled divided by the airspeed in each segment. As it has been stated when the angle change between two segments is lower than $\pi/8$ rad, no turn is considered and the distance is the value of the straight segment. However, if the angle is higher, the trajectory travelled is similar to the

response of corresponding transfer function. Figure 3 shows an example when α_{ijk} is $\pi/2$ rad; the trajectory described is composed of two sub-trajectories. On one hand, l_a follows the dynamics of the derivate of the step response until the values remains within ±0.5 km. This value has been fixed for all the transfer functions for the purpose of disregard the final oscillations of the responses (Fig. 2). On the other hand, once the response stays within ±0.5 km, the trajectory is contemplated as a straight line until the end of the evaluated segment, which is the sub-trajectory designated as l_b. So, the time taken objective function for each arc of the grid map can be defined as:

$$\begin{aligned} &if\ 0 \le \alpha_{ijk} < \pi/8 \quad t_{ij} = d_{ij}/V \\ &if\ \pi/8 \le \alpha_{ijk} \le \pi \quad t_{ij} = (l_a + l_b)_{ij}/V \end{aligned} \tag{5}$$

- Fuel consumption, *fc* (kg): The fuel consumption is described as the addition of the fuel burned along the route. The consumed fuel in each section of the path can be calculated as:

$$\begin{aligned} &if\ 0 \le \alpha_{ijk} < \pi/8 \quad fc_{ij} = fb \cdot t_{ij} \\ &if\ \pi/8 \le \alpha_{ijk} \le \pi \quad t_{ij} = \left[l_a \cdot (fb + pc) + l_b \cdot fb\right]_{ij}/V \end{aligned} \tag{6}$$

  As it can be seen, it depends on the aforementioned lengths (l_a and l_b) (Fig. 3), the constant fuel burn *fb* and airspeed *V*, the time taken to travel that section t_{ij} and a penalty coefficient *pc*. The penalty coefficient is obtained by multiplying by 25 the number of sign changes of the curve l_a. The objective of this coefficient is to add a variable that approximately values the extra fuel needed when turning. This approximation is purely symbolic and has not been set according to real data, since, as mentioned previously, the aircraft dynamics has been simplified.
- Area deviation, A_d (km^2): The area deviation as objective function is the summation of all the areas covered by the curve la in each section of the path (Fig. 3). This means that, if α_{ijk} is lower than $\pi/8$ rad, the area in segment (n_j, n_k) is zero because the trajectory has been considered as the straight line that connects n_j and n_k.

IV. Heuristic Cost Calculation

The evaluation function, objective function or node-selection function vector $\vec{f}(n)$, that NAMOA* uses to find the optimal paths, consists of adding two vectors: $\vec{g}(n)$ which is the cost of the current path from n_s to n, and $\vec{h}(n)$ called heuristic function vector, which denotes an estimate of the cheapest cost from n to n_g [9]. Several methods were used to calculate the heuristic costs vector estimations. The performances of the methodologies were compared between each other for different maps to decide which one was the best to be finally used with NAMOA* algorithm.

A. Methodologies to Calculate Heuristic Costs

- "One step" method: With this method, first, it was calculated and saved the costs vector from the current node *n* to each of its successors, without including its predecessor. Then, the heuristic cost vector of *n* was made equal to the non-dominated cost vector from those saved. If there was more than one non-dominated vector, it was chosen the first saved.
- "All steps" method: In this case, it was calculated the costs vector of the path from the current node *n* to the goal node through intermediate nodes. This methodology added the cost vectors of each arc until it reached the goal. After that, the method only recorded those non-dominated cost vectors and, as the previous method, the heuristic vector was made equal to the non-dominated one or to the first non-dominated if there were several. As it can be deduced, the computational requirements of this method were quite high because it practically explored the entire grid map to obtain the vector values. In the next section, it will be analyse how the computational time increased as the number of nodes in a grid rose. It also has to be pointed out that, this way of obtaining the heuristic cost vector was, itself, an inefficient heuristic optimisation search problem, because in the first iteration, when the heuristic vector of the start node was calculated, this method came out with the final solution of the search problem, that is, the optimal cost of going from the start to the goal node; and despite this, it continued calculating the heuristic vector of other nodes in the next iterations.
- "Direct" method: In this method, the heuristic cost vector is the cost vector of an arc that joins, in a straight line, node *n* with the goal node. The fact that it is probable that those nodes are not graphically connected, does not affect to the calculation.

B. Performance and Comparison of the Methodologies Used to Calculate Heuristic Costs

NAMOA* code were run for five different grid maps, each map more complex than the previous ("MAP 1" to "MAP 5"), for the three methodologies developed to calculate the heuristic cost vectors of a multi-

objective problem in which distance travelled and angle change were the objective functions. Then, three parameters were compared: in the first place, the result of the optimisation problem, that is, the non-dominated cost vectors together with corresponding paths; secondly, the time that takes the algorithm to get the results and thirdly, the number of evaluations done by NAMOA*, considering one evaluation as each calculation of the evaluation function vector $\vec{f}(n)$.

The resultant optimal cost vectors and paths were the same for each map independently of the methodology used which demonstrates compliance with NAMOA* admissibility property [9].

TABLE I. PERFORMANCES OF THREE METHODOLOGIES USED TO CALCULATE HEURISTIC COST VECTORS.

	Directed grid map	"One step" method		"All steps" method		"Direct" method	
		Runtime [*s*]	No. eval	Runtime [*s*]	No. eval	Runtime [*s*]	No. eval
MAP 1	Yes	0.0869	12	0.0758	9	0.0674	9
	No	0.0906	14	0.0872	11	0.0689	11
MAP 2	Yes	0.0992	16	0.0735	9	0.0661	9
	No	0.0984	18	0.0898	9	0.0652	9
MAP 3	Yes	0.1089	22	0.0795	10	0.0746	17
	No	0.1224	29	0.1432	10	0.0807	21
MAP 4	Yes	0.1408	36	0.0884	12	0.0923	23
	No	0.1503	45	0.4206	14	0.0840	29
MAP 5	Yes	0.6036	237	6.6955	48	0.1188	100
	No	2.8039	411	-	-	0.1323	155

Table I presents the runtime and the number of evaluations of NAMOA* code. As it was expected and is manifested in the table, as the size of maps increased, these two variables also rose. Such behaviour was common to three methods. For scenarios with small number of nodes ("MAP 1" to "MAP 3"), runtime and evaluations hardly changed for directed grids with respect to undirected ones, for all methods. However, for bigger graphs ("MAP 4" and "MAP 5"), changes began to manifest more clearly. Comparing the data of the last two maps for all methodologies, it can be seen that "All steps" is the one with higher time values and lower evaluations. This is an indicative of characteristics of this method, mentioned in the previous section. On one hand, the computational complexity required to explore nearly the entire graph to calculate a single heuristic vector is manifested by the time needed. On the other hand, the number of evaluations is lower because from the beginning, "All steps" method came out with an optimal cost, thus, the algorithm seems to prune earlier irrelevant nodes which will not be evaluated; in addition, the evaluations performed by the function that calculates the heuristic function has not been added to NAMOA* evaluations to make an equitable comparison with the other methods. If those evaluations had been counted, they would reflect that the high runtime needed is caused by them. Strong evidence of the computational requirements is found for undirected "MAP 5" data; the absence of data for the second method is due to the long time took by the code to be run, which made not possible to record it. This behaviour determined the decision to reject its implementation in this study.

A comparison of "One step" and "Direct" results, from table I, reveals the advantage of "Direct" over "One step" methodology, with lower number of evaluations and lower times, due to "Direct" method used the goal position as a reference to calculate the heuristic function while "One step" did not make use of that information. This analysis indicates that "Direct" method was the most suitable for the posed problem of this research. Moreover, it was desirable to use scenarios in with the graphs were undirected and quite big, like the civil airspace, and "Direct" method showed the best performance for them.

V. MULTI OBJECTIVE OPTIMISATION PROBLEM

This study set out with the aim of implementing time taken, fuel consumption and area deviation as the main objective functions of a multi-objective problem in scenarios that simulate the prohibited zones and time and speed constraints of civil flyable space. Four scenarios were created to show the performance of NAMOA*.

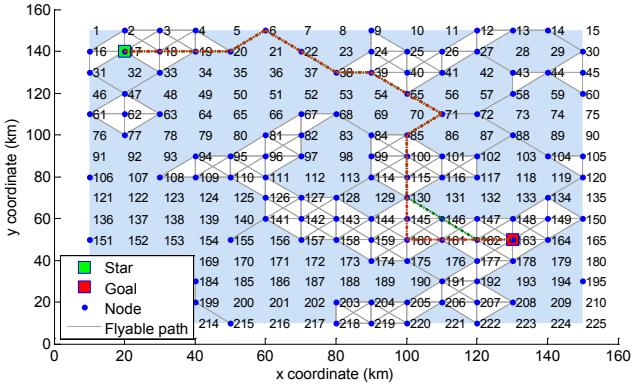

Fig. 4. Optimal paths (dash-dotted coloured lines) (see II) for a three objective (time, fuel and area deviation) NAMOA* optimisation of an undirected grid map with blue coloured prohibited zones (scenario 1).

TABLE II. OPTIMAL PATHS AND ITS CORRESPONDING COST VECTORS FOR THE MAP SHOWN IN FIGURE 4

Optimal path	Non-dominated cost vector (time [*s*], fuel [*kg*], area [km^2])
(n_{17}, n_{18}, n_{19}, n_{20}, n_6, n_{22}, n_{38}, n_{39}, n_{55}, n_{71}, n_{85}, n_{100}, n_{115}, n_{130}, n_{146}, n_{162}, n_{163})	(2.1500, 0.1528, 0.0084) $\cdot 10^3$
(n_{17}, n_{18}, n_{19}, n_{20}, n_6, n_{22}, n_{38}, n_{39}, n_{55}, n_{71}, n_{85}, n_{100}, n_{115}, n_{130}, n_{145}, n_{160}, n_{161}, n_{162}, n_{163})	(2.2721, 0.1614, 0.0078) $\cdot 10^3$

The first scenario (Fig. 4) is a grid map with flyable nodes connected between each other and represented, in all the figures of this section, by blue dots. This particular scenario has some nodes that are considered obstacles and are encompassed in a blue coloured area by way of danger or prohibited areas of airspace. Two optimal paths were found (Fig. 4) with different and non-dominated cost vectors (Table II). The decision on which path to choose would be the responsibility of other agents. This decision is not part of the objectives of this research, but gives rise to the development of future studies.

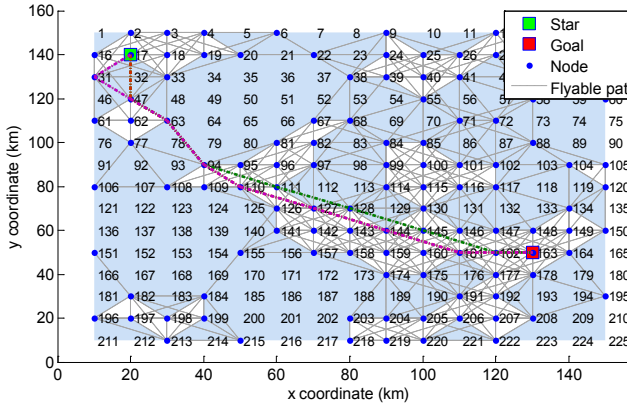

Fig. 5. Optimal paths (dash-dotted coloured lines) (see table III) for a three objective (time, fuel and area deviation) NAMOA* optimisation of an undirected grid map with blue coloured prohibited zones and more connections representing "safe corridors" (scenario 2).

TABLE III. OPTIMAL PATHS AND ITS CORRESPONDING COST VECTORS FOR THE MAP SHOWN IN FIGURE 5.

Optimal path	Non-dominated cost vector (time [s], fuel [kg], area [km^2])
(n_{17}, n_{47}, n_{63}, n_{94}, n_{111}, n_{128}, n_{145}, n_{162}, n_{163})	(1.7059, 0.1210, 0.0027) $\cdot 10^3$
(n_{17}, n_{47}, n_{63}, n_{94}, n_{110}, n_{127}, n_{144}, n_{161}, n_{163})	(1.7183, 0.1218, 0.0018) $\cdot 10^3$
(n_{17}, n_{47}, n_{63}, n_{94}, n_{110}, n_{127}, n_{144}, n_{161}, n_{162},	(1.7183, 0.1218, 0.0018) $\cdot 10^3$

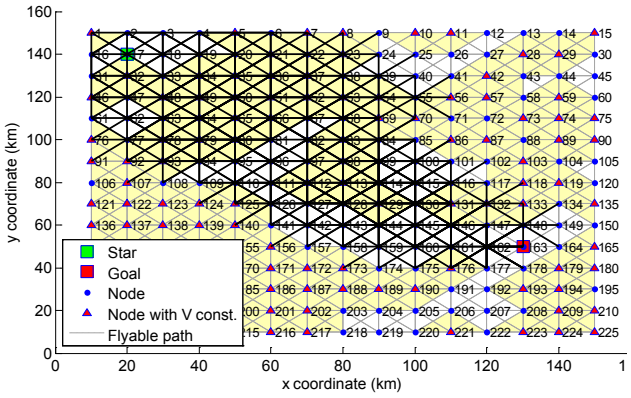

Fig. 6. Paths evaluated (solid black lines) during a three objective (time, fuel and area deviation) NAMOA* optimisation of an undirected grid map with yellow coloured areas encompassing nodes with speed constraints (scenario 3).

The second scenario (Fig. 5) is the same as the previous one with the difference that there are more flyable paths between nodes. Those extra arcs represent "safe corridors" which can be negotiated with airspace controllers. The approach for future researches is to compare the cost advantages of considering or not those "corridors" and accordingly evaluate which ones should be negotiated.

Comparing last two tables, it can be noticed that the use of "safe corridors" generates paths (Table III) with costs that dominate the costs of those paths obtained without considering "corridors" (Table II).

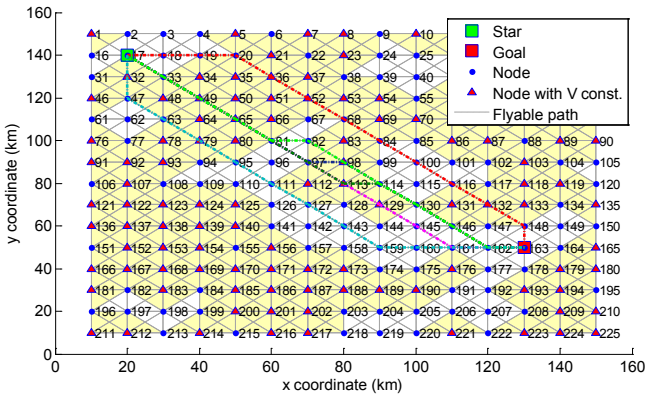

Fig. 7. Optimal paths (dash-dotted coloured lines) (see table IV) for a three objective (time, fuel and area deviation) NAMOA* optimisation of an undirected grid map with yellow coloured areas encompassing nodes with speed constraints (scenario 3).

TABLE IV. OPTIMAL PATHS AND ITS CORRESPONDING COST VECTORS FOR THE MAP SHOWN IN FIGURE 7.

Optimal path	Non-dominated cost vector (time [s], fuel [kg], area [km^2])
(n_{17}, n_{33}, n_{49}, n_{65}, n_{81}, n_{97}, n_{113}, n_{129}, n_{145}, n_{161}, n_{162}, n_{163})	(1.9047, 0.1350, 0.0010) $\cdot 10^3$
(n_{17}, n_{33}, n_{49}, n_{65}, n_{81}, n_{97}, n_{98}, n_{114}, n_{130}, n_{146}, n_{162}, n_{163})	(1.8244, 0.1294, 0.0029) $\cdot 10^3$
(n_{17}, n_{33}, n_{49}, n_{65}, n_{81}, n_{97}, n_{113}, n_{114}, n_{130}, n_{146}, n_{162}, n_{163})	(1.8426, 0.1307, 0.0029) $\cdot 10^3$
(n_{17}, n_{33}, n_{49}, n_{65}, n_{81}, n_{82}, n_{98}, n_{114}, n_{130}, n_{146}, n_{162}, n_{163})	(1.8468, 0.1310, 0.0029) $\cdot 10^3$
(n_{17}, n_{18}, n_{19}, n_{20}, n_{36}, n_{52}, n_{68}, n_{84}, n_{100}, n_{116}, n_{132}, n_{148}, n_{163})	(1.9029, 0.1349, 0.0019) $\cdot 10^3$
(n_{17}, n_{32}, n_{47}, n_{63}, n_{79}, n_{95}, n_{111}, n_{127}, n_{143}, n_{159}, n_{160}, n_{161}, n_{162}, n_{163})	(1.8633, 0.1321, 0.0019) $\cdot 10^3$

A third scenario was tested. What in previous scenarios were considered obstacle nodes, now are flyable nodes with speed restrictions (red triangles in Fig. 6 and Fig. 7). As it was previously stated, airspeed (V) was assumed constant, with a value of 92.60 m/s for all nodes except for those in a speed constraints area, in which the value is 61.73 m/s. The speed constraints area or the area which encompasses nodes with speed constraints has been coloured yellow in figures. Figure 6 provides the visualisation of the paths evaluated by the algorithm, of which spread is more notable at the beginning. When NAMOA* found the first solution,

nodes expansion started to be pruned, keeping the evaluations tighter to final paths.

Again, Fig. 7 and table IV presents the results reached with NAMOA* optimisation code. As expected, in the absence of obstacles, the paths are nearly straight and attempt to go through the fewest number triangles as possible. It has to bear in mind that the values shown in tables have a limited number of significant figures. For example, in table IV the third cost vector might look dominated by the second, but in fact the third component of the third vector is lower than that corresponding to the second vector.

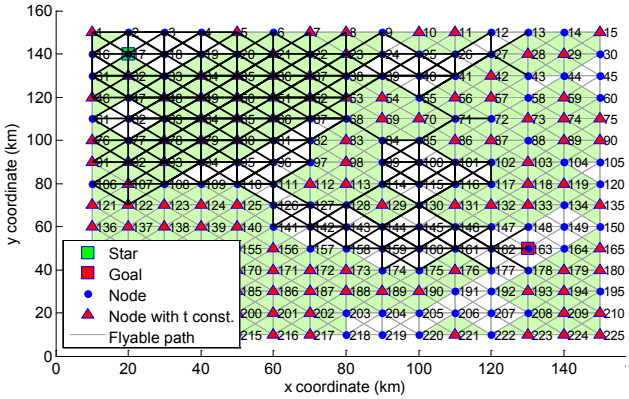

Fig. 8. Paths evaluated (solid black lines) during a three objective (time, fuel and area deviation) NAMOA* optimisation of an undirected grid map with green coloured areas encompassing nodes with time constraints (scenario 4).

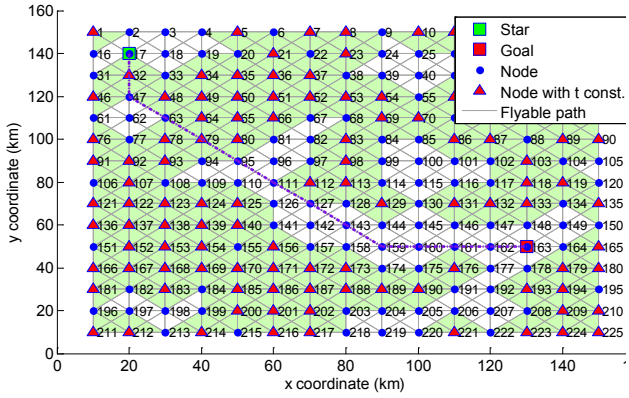

Fig. 9. Optimal paths (dash-dotted coloured lines) (see table V) for a three objective (time, fuel and area deviation) NAMOA* optimisation of an undirected grid map with green coloured areas encompassing nodes with time constraints (scenario 4).

The last scenario (Fig. 8 and Fig. 9) is equivalent to the previous one. The difference is that nodes marked by a triangle, have time constraints instead of speed constraints. This means that, if a triangular node was reached at a time higher than a previously known time limit, that node was not considered for consecutive path expansions. The time limit was established to 800 seconds and it was compared with the first objective function, time taken to travel to a node.

TABLE V. OPTIMAL PATHS AND ITS CORRESPONDING COST VECTORS FOR THE MAP SHOWN IN FIGURE 9

Optimal path	Non-dominated cost vector (time [s], fuel [kg], area [km^2])
(n_{17}, n_{32}, n_{47}, n_{63}, n_{79}, n_{95}, n_{111}, n_{127}, n_{143}, n_{159}, n_{160}, n_{161}, n_{162}, n_{163})	$(1.7328, 0.1229, 0.0019) \cdot 10^3$

Once explained that, it is easier to understand the behaviour of the evaluated paths in Fig. 8. It can be noticed how, close to the start node, arcs that connect nodes with time constraints were evaluated until at certain area in which the green coloured zone was no longer considered to evaluation. That invisible line which delimits the evaluated green area and the not evaluated was caused by the time limit which blocked the expansion to those nodes.

Finally, Fig. 9 shows the only optimal path found in this fourth scenario, of which cost vector is detailed in Table V. This single cost vector indicates that it is a solution which minimises all the objectives and the rest of possible solutions are dominated by it. It is interesting to notice that the path of this table is the same as the last one of table IV, although the cost vectors are not equal due to the different speed values used to calculate them in those two last scenarios. Because of this coincidence, it can be deduced that if a scenario had combined scenario 3 and 4, the mentioned path would have been, at least, one of the results.

VI. HEURISTIC MULTI-OBJECTIVE SEARCH COMPARISON

In order to validate NAMOA* performance, it was compared with another algorithm as it is weighted A*. Weighted A* applies Weighted Sum Method (WSM) [14] to unify multiple objectives values into a single one, by multiplying each objective function by a weight and adding them, and then, uses that scalar cost value in A* algorithm. A total of 100 weight vectors were used for each case of study. When it was a two objective case the vectors took values from (1, 0) to (0, 1), and when it was a three objective case the three components of the 100 vectors were generated randomly for values between 0 and 1. For both of them the summation of the components of each weight vector was not higher than 1.

The comparison was done using the Inverted Generational Distance (IGD) [15] between weighted A* and NAMOA* algorithm's set of solutions and the estimated Pareto-optimal front. This front was obtained by combining all the results found with both algorithms, separately for each scenario and group of objective functions, and choosing those non-dominated. Therefore, the Pareto-optimal front is a reference and

the best known estimation of the optimal front; however, it can not be ensured that is the optimal one.

TABLE VI. THE IGD VALUES FOR WEIGHTED A* AND NAMOA* APPLIED TO ALL SCENARIOS AND THE OBJECTIVE FUNCTIONS INDICATED.

Scenario	Objective functions	IGD (weighted A*)	IGD (NAMOA*)
1 (prohibited zones)	Distance and angle	0.0000	0.0000
	Distance, angle and no. odd	14.2694	0.0000
	Time and area	16.2825	0.0000
	Time, fuel and area	61.2020	0.0000
2 (prohibited zones and "safe corridors")	Distance and angle	0.0000	0.0000
	Distance, angle and no. odd	0.4278	0.0000
	Time and area	4.4936	0.0000
	Time, fuel and area	4.5035	0.0000
3 (speed constraints)	Time and area	3.5339	0.0000
	Time, fuel and area	14.8965	0.0000
4 (time constraints)	Time and area	7.2134	0.0000
	Time, fuel and area	7.2358	0.0000

Table VI presents the IGD values for weighted A* and NAMOA* algorithms for each case analysed. The performance of NAMOA* was clearly the best with a set of solutions equal to the Pareto-optimal front, which was represented by zero IGD values. However, some weighted A* solutions were not in that front which was reflected by nonzero IGD values. Weighted A* only generated zero IGD values for distance and angle objectives in scenario 1 and 2 because some, but not all, of its solutions coincided with the Pareto front. Nevertheless, the small amount of solutions, for some cases of study, requires more tests in different scenarios to be able to draw a definitive conclusion from them.

Apart from this, it is interesting to notice how for weighted A*, the IGD values were higher when it was added a third objective to a two objective problem. This might has been due to the randomly generation of the weight components which values may not cover the whole range between 0 and 1.

TABLE VII. NUMBER OF EVALUATIONS PERFORMED BY WEIGHTED A*(NOT USING AND USING SAVED EVALUATION FUNCTIONS COSTS FROM PREVIOUS EVALUATIONS) AND NAMOA* WHEN THEY WERE APPLIED TO ALL SCENARIOS AND THE OBJECTIVE FUNCTIONS INDICATED.

Scenario	Objective functions	Weighted A* No. eval.		NAMOA * No. eval.
		Not saving $f(x)$	Saving $f(x)$	
1 (prohibited zones)	Distance and angle	6343	76	177
	Distance, angle and no. odd	6303	93	416
	Time and area	6431	84	209
	Time, fuel and area	6402	65	209
2 (prohibited zones and "safe corridors")	Distance and angle	6897	104	445
	Distance, angle and no. odd	7624	189	1086
	Time and area	7611	130	474
	Time, fuel and area	7311	92	474
3 (speed constraints)	Time and area	13462	174	880
	Time, fuel and area	13463	148	880
4 (time constraints)	Time and area	13230	184	542
	Time, fuel and area	13155	135	542

Table VII provides a comparison of the number of evaluations made by weighted A* saving or not prior information and NAMOA*. The evaluations for each weighted A* case of study is the summation of the evaluations done for the 100 independent runs with different weights, however, in the second column, A* was modified to save the evaluation function vectors to avoid re-evaluations in the total 100 runs. The number of evaluations of NAMOA* is the value for a single run. Three significant features can be extracted from this table. Firstly, it is extremely notable how NAMOA* performs fewer evaluations than weighted A* without prior information, but, higher evaluations than weighted A* using that information. Secondly, the already mentioned randomly generation of the weight components for three objective cases with weighted A* might be the reason of the, in general, lower number of evaluations of weighted A* without prior data, when there was three objectives than when there were two. And thirdly the repeated number of NAMOA* evaluations when the objectives are time and area, and time, fuel and area, backs up the finding of a linear relation between time and fuel, that is, when the time is minimised the fuel consumed is also minimised. However, there does not exist a linear relation in Eq. 5 and Eq. 6 when the angle between arcs change more than $\pi/8$, but it can not be said the same for lower angles. This finding might be caused by the characteristics of the maps used and it may be probable that for maps with arcs less symmetrical and with changes in the angles higher than $\pi/8$, this relation between these objectives would not be manifested. This means that the optimal paths of the three objective problems were equal to the optimal paths of a two objective problems in which either time or fuel is not considered.

VII. CONCLUSIONS AND FUTURE WORK

This paper has presented the functionality of NAMOA* multi-objective optimisation graph-search algorithm to automatically plan a route for a simulated UAV in civil airspace taking into account simultaneous and conflicting objectives that must be analysed to find the set of optimal paths to go through, from the starting point to the known single final point.

The results have shown, NAMOA* is a good multi-objective algorithm that in a single run, finds a set of solutions equal to the estimated Pareto-optimal set for several scenarios configured with directed or undirected arcs and different constraints. However, a fast

performance in terms of runtime and number of evaluations mainly depends on the methodology used to calculate the heuristic function vector and the complexity of the objective functions.

NAMOA* has shown to be superior to weighed A* in the sense that in a single run is able to obtain a set of non-dominated solutions while weighted A* will only be able to achieve the same results if several weight vectors are previously carefully tuned. So, although weighted A* might be able to perform the search in lower evaluations, the effort needed to calculate the weight combinations for each specific case, dismissed its use for the problem formulated in this study, in which the objective is to obtain all the possible non-dominated solutions to, subsequently, provide them to a decision making agent. This agent could be a person in a ground control station or the vehicle itself, who or which will value if, for instance, the solution with lower fuel consumption should be a priority over others. Future work should be focused on this decision making process as well as study, taking into account the aerial regulations, the great potential of negotiate "safe corridors" through which an UAV will be able to cross certain restricted areas. Also, future work is needed to deal with online route planning that builds maps as the UAV moves forward.

REFERENCES

[1] BAE Systems, (2013), *Pioneering first remotely piloted flight in UK airspace*, available at: http://www.baesystems.com/article/BAES_158030/pioneering-first-remotely-piloted-flight-in-uk-airspace (accessed 23th October 2013).

[2] Pohl, A.J. and Lamont,G.B., (2008), "Multi-objective UAV mission planning using evolutionary computation", in: *Proceedings of the 40th Conference on Winter Simulation*, 7-10 December 2008, Miami, FL, USA, Edited by Mason, Hill, Mönch, Rose, Jefferson and Fowler, pp. 1268-1279.

[3] Dijkstra, E. W., (1959), "A note on two problems in connexion with graphs", *Numerische Mathematik*, Vol. 1, No. 1, pp. 269–271.

[4] Hart, P.E., Nilsson, N.J. and Raphael, B., (1968), "A formal basis for heuristic determination of minimum path cost", *IEEE Transactions on Systems, Science, and Cybernetics*, Vol. 4, No. 2, pp. 100–107.

[5] Leedy, B., Putney, J., Bauman, C., Cacciola, S., Webster, J. and Reinholtz C., (2006), "Virginia Tech's Twin Contenders: a comparative study of reactive and deliverative navigation", *Journal of Field Robotics*, Vol. 23, No. 9, pp. 709-727.

[6] Hansen, P., (1979), "Bicriterion path problems", in: *Proceedings of the Third Conference of Multiple Criteria Decision Making Theory and Application,* Vol. 177, 20-24 August 1979, Hagen/Königswinter, Germany, Springer, Berlin, pp. 109–127.

[7] Stewart, B. S. and White, C. C., (1991), "Multiobjective A*", *Journal of the ACM*, Vol. 38, No. 4, New York, USA, pp. 775–814.

[8] Mandow, L. and Pérez de la Cruz, J.L., (2005), "A new approach to multiobjective A* search", in: *Proceedings of the XIX International Joint Conference on Artificial Intelligence (IJCAI-05)*, 31 July - 5 August 2005, Edinburgh, Scotland, UK, Professional Book Center, pp. 218-223.

[9] Mandow, L. and Pérez de la Cruz, J.L., (2010), "Multiobjective A* search with consistent heuristics", *JACM*, Vol 57, No. 5, Article 27.

[10] Machuca Sánchez, E. L., (2012), *An analysis of some algorithms and heuristics for multiobjective graph search* (unpublished PhD thesis), Universidad de Málaga, Málaga, Spain.

[11] Waldock, A. and Corne, D. (2012), "Exploiting prior information in multiobjective route planning", in: *Proceedings of the 12th International Conference in Parallel Problem Solving from Nature, Part II* , Vol. 7492, Taormina, Italy, 1-5 September 2012, Springer, Berlin, pp. 11–21.

[12] Coughlan, W., (2006), *Development and validation of a large amplitude simulation of the Cranfield Jetstream 31*, (unpublished MSc thesis), Cranfield University, Cranfield.

[13] Cooke A. K., (2013), "J31 (G-NFLA) flight test report-work book", Lecture notes given to support the Group Flight Test Report (Introduction to Aerodynamics, Aircraft Performance, Stability and Control) course at Cranfield University, 2013, unpublished.

[14] Marler, R.T. and Arora, J.S., (2010), "The weighted sum method for multi-objective optimization: new insights", *Structural and Multidisciplinary Optimization*, Vol. 41, No. 6, pp. 852-862.

[15] Schutze, O., Esquivel, X., Lara, A. and Coello Coello, C.A., (2012), "Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization", IEEE Transactions on Evolutionary Computation, Vol. 16, No. 4, pp. 504 – 522.