

REMOTELY PILOTED AIRCRAFT ROUTE OPTIMIZATION WHEN PERFORMING OIL POLLUTION MONITORING OF THE SEA AQUATORIUM.

Aleksandrs URBAHS, Vladislavs ŽAVTKĒVIČS

*Institute of Aeronautics, Riga Technical University, Kalku str. 1A, k-1, LV-1658 Riga, Latvia
E-mail: aerti@rtu.lv (corresponding author)*

Received 30 January 2017; accepted 15 June 2017



Aleksandrs URBAHS, Professor, Dr Habil. Sc. Eng.

Education: 1981 – Diploma of Mechanical Engineer, Riga Civil Aviation Engineering Institute, Faculty of Mechanical Engineering

1986 – Dr sc.ing., Riga Civil Aviation Engineering Institute, Faculty of Mechanical Engineering

1997 – Dr hab. sc.ing., Riga Aviation University.

Affiliations and functions: 1989–1999 – Vice Dean and Dean, Faculty of Mechanical Engineering, Riga Aviation University. Since 1999 – Full time Professor, Riga Technical University.

1999–2012 – Director of the Institute of Transport Vehicles Technologies of Riga Technical University.

Since 2012 – Director of the Institute of Aeronautics of Riga Technical University.

Research interest: Aeronautics, Transport and Mechanical Engineering, Unmanned Vehicles, Nano-Materials, Non destructive methods of control.

Publications: 326 scientific papers and 41 patents.

Present position: Professor, Director of the Institute of Aeronautics of Riga Technical University, Expert at Latvian Council of Science ect.

Research interests: Transport and traffic engineering, air transport and infrastructure, remote piloted aircraft technologies.



Vladislavs ŽAVTKĒVIČS has been a Doctoral Student at the Institute of Aeronautics, Riga Technical University since 2013. In 2010 he obtained a degree of Master of Science at the Latvian Maritime Academy. He graduated from the Latvian Maritime Academy in 2004 and obtained specialisation Engineer – Ship's Navigator.

Research interests: Transport and traffic engineering, air transport and infrastructure, remote piloted aircraft technologies.

Abstract. The main problem of sea aquatorium monitoring is the surveillance of large areas of the sea surface and the presence of many moving ships with different parameters. A methodology for the optimization of a Remotely Piloted Aircraft's (RPA) route is presented.

Keywords: algorithm, mission planning, optimization.

1. Introduction

The main problem of sea aquatorium monitoring is the surveillance of large areas of the sea surface and the presence of many moving ships with different

parameters. The situation with ship traffic is changing dynamically, and the main objective is to develop an optimization algorithm for an RPA mowing between the targets (ships) and the probable oil pollution areas.

Linear programming methods can be implemented only for a small number of static objects.

A typical monitoring mission involves planning the RPA flight in accordance with many requirements. These requirements are necessary to monitor the ships moving through the sea and their tracks that have been dynamically changed. The main challenge is the implementation of the RPA for a mission with both static and dynamic objects. The ground control station operator who is selecting an RPA with a dedicated set of sensors for oil pollution monitoring mission should take into account the optimization of the flight plan. The operator of a manually controlled RPA should solve the drift of the RPA caused by the speed of the wind, the probable trajectory of moving targets, depending on the target as well as the RPA courses and target parameters. Current technologies using software applications allow transferring control from the operator to the autonomous system. A dynamical programming algorithm can perform the RPA route optimization for a distance matrix between targets. The main approach for finding an optimal route between points of interest is solving the travelling problem with Dijkstra's algorithm. Dijkstra's algorithm includes several heuristics to direct the vehicle towards the goal state and help to eliminate unnecessary computations by removing an unnecessary state analysis (Dijkstra 1959).

2. Optimization of the surveillance mission

For the representation of the dynamical motion of targets and possible moving oil slicks a map with calculated target data probabilistic positions received from an Automatic Identification System (AIS) can be used. Mission planning with the integration of information from the AIS system and the shore radar surveillance systems will allow minimizing the flight time of the monitoring mission and the RPA operational cost. To formulate the problem one needs a set of targets with specific parameters, such as the type of the target, an equation of motion, the last received positions from the sea traffic monitoring systems, and the formulation of the presentation (point, line, and area). For the optimization of surveillance missions of oil pollution monitoring of sea objects performed by the RPA, a criterion that must be taken into account is the visiting of the targets matrix. The next stage includes the methodology for planning oil pollution and targets that can require oil pollution monitoring. The targets of interest will be divided into classes: high pollution risk (oil/chemical tankers and ships' routes), medium risk (cargo ships), and small risk targets (fishing vessels). The planning of oil pollution surveillance missions includes a great number of possible decisions. The planning of a successful mission of the RPA should be designed as a dynamic system with feedback from the information about the RPA and the targets' positions. Dynamical programming methods

allow solving different parts of the problem (subproblems), and then combining the solutions of the subproblems to reach an overall solution. There are many types of algorithms for using the RPA for monitoring and surveillance tasks of static targets with speed $V = 0$. While the monitoring of static targets with one or more moving objects is performed, the decision and results of these algorithms will not be optimal for the defined problem with dynamic objects. The target of the dynamical environment can be represented as points with such main parameters as initial speed and course. The targets are divided into four groups with speed $V = 0$, $0 < V < 14$, $14 < V < 23$, and $23 < V$ (in knots). If a coordinate system which includes measuring the distance between the RPA and the objects of surveillance is created, good results will be provided. However, it is very important to develop an automatic data calculating and collecting system performing the calculation of the relative motion of the RPA taking into account the effect of the wind speed. In this approach the relative motion of both types of targets, dynamical and static, is represented on the plane by two equations $X = \text{RPA}N - \text{TN}$, $Y = \text{RPA}E - \text{TE}$, where TN and TE are the Cartesian coordinates of the target on the plane. In dynamical programming methods, the optimization of decision finding will be performed into two stages. Taking into account the motion parameters of the RPA (x_0, y_0, V_0), the parameters of objects with V , and the course, the distance between objects and the RPA will be found. To find an optimization solution for the RPA route for dedicated tasks, the targets from the electronic AIS map, the coastal radar surveillance systems, and the sea surface areas will be presented as a graph with nodes and edges with weights. The decision graph represents all possible routes between all types of objects that should be surveyed during the mission (Guerin, Orda 2002). The nodes of the graph are different objects and targets. The objects and targets in the graph should be divided into the following classes: point, linear, static, and specific area objects.

The weight of the graph edges is the mission cost for each object. The cost in the dynamic system is presented by the time function (Stipanović *et al.* 2004). To determine the weight of edge, the following formula is used: $w_{ij} = t_{flightij} + t_{surveillance}$, where $t_{flightij}$ is the flight time from the previous object, and $t_{surveillance}$ is the duration of the object surveillance mission. According to the classification, the duration of the surveillance of all objects should be calculated using the following formula:

$t_{surveillance} = \frac{l_{trajectory}}{V_{flight}}$, where $l_{trajectory}$ is the length of the surveillance trajectory and V_{flight} is the speed of the RPA in the surveillance mode. The task of the surveillance mission can be represented as the process of visiting all nodes on graph $G(V, E)$ starting from the start point

$S \in V[G]$ and returning to the start point. The objective is to find the optimal route for the RPA, the aim of which is to visit all objects recorded in the task in the shortest possible time and with the least cost. In graph theory it is searching for ways of linking two or more nodes using optimality criteria (Hung, Divoky 1988). The tree building algorithm starts with a tree with one vertex. During the working process of Prim's algorithm, the tree will include all vertexes of the RP mission description graph. At each step the tree adds a slight edge connecting the tree and a separate vertex from the rest of the graph. This rule only adds safe edges; therefore, at the end of the algorithm, the edges build a minimum spanning tree. The tree algorithm is related to heuristic methods. In heuristic methods to select elements for the solution, it is recommended to use some natural selection rules. The time of this algorithm is equal to $\Theta(E) = \Theta(V^2)$.

In dynamic programming methods, the comparison of all possible problem solutions is not carried out at the end of the construction of the set of all possible options, but at every step of the sequence of option construction (Bellman 1957). To solve the problem, T should be defined as the sequence of subtasks T1, T2, T3, Tn, that task T1 already has a solution $T = T_n$, and solutions of the tasks T1, T2 and Ti-1 are known, and can be derived from the solution of the problem Ti for all $i = 2 \dots n$. The procedure of finding the optimal control (the decision) is based on the analysis of the recurrence relation.

$$f_{n-1}(S_l) = \text{optimum}[R_{l+1}(S_l, U_{l+1}) + f_{n-(l+1)}(S_{l+1})],$$

$l = \overline{0, n-1}$. Where $U_l = (u_1^l, \dots, u_m^l)$ is the solution chosen at the l-st step; $S_l = (s_1^l, \dots, s_m^l)$ is the state of the system at the l-st step; R_l – the immediate effect achieved at the l-st step; f_{n-l} – the optimal value of the effect achieved for the n-l steps; n – the number of the steps (stages). The sea targets for the surveillance mission performed by the RPA implementing the dynamical programming principle can be presented by the following example. Suppose, that the shortest flight route from ship A to ship C is known as f^* and passes through ship B, state i. The optimality decision principle states that the route leg from B to C has the shortest distance and the minimal cost, and it starts in B and finishes in C, f^*i (Fig. 1).

The motion of the object, for example, a ship, is solved by using the parameters of the moving object. In this case it is maintained that the target is moving with a constant course and speed. This methodology gives an opportunity to find a meeting point where the RPA reaches the moving object. To determine the meeting point of the moving object the following functions are used: the function that determines the time of the movement of the object to one of the points on the straight line along which the object is moving; the function that determines the movement of the RPA to the point

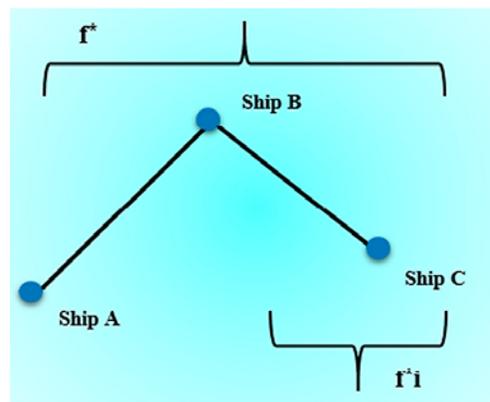


Fig. 1. Dynamical programming principle

mentioned above. To represent the moving object, the traverse sailing method is used. This method allows computing the position of the object as well as the course and distance to the next point. In this case, the distance of the sea target from the starting point is calculated using the following formula: $D = V \times t$ where V is the speed of a ship in knots from the AIS transponder information, t is time. For point P2 (ϕ_2, λ_2) latitude ϕ_2 is determined by the formula $\phi_2 = \phi_1 + V \times t \times \cos \text{Course}$, where ϕ_1 is the latitude of the starting point p1, the course of a ship is obtained from the AIS transponder information. The longitude of point P2 is determined by the formula

$$\lambda_2 = \lambda_1 + V \times t \times \sin \text{Course} \times \frac{1}{\cos \frac{\phi_1 + \phi_2}{2}}.$$

The main principle of controlling the RPA model in vector form is $\dot{x} = f(x, u, t)$, where $t \in [t_0, t_k]$, and $x(t_0) = x_0$. In this equation $x = (x_1, x_2, \dots, x_n)$ is the vector of the RPA condition (the coordinate vector), $fu = (u_1, u_2, \dots, u_m)$ – the controlling vector, $f = (f_1, f_2, \dots, f_n)$ – the vector function of their arguments, $[t_0, t_k]$ – the time interval during which an RPA flight is performed. To control the RPA during the mission planning, the software should take into account the restriction $u_{\min} \leq u(t) \leq u_{\max}$, where $t \in [t_0, t_k]$ (Fig. 2).

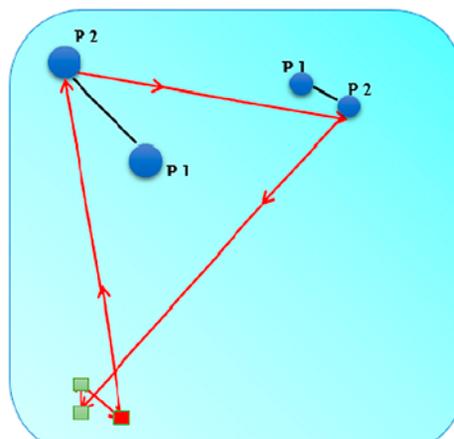


Fig. 2. Representation of moving objects

To solve this problem, the efficiency of obtaining data about the object of observation is very important. There are several algorithms for constructing the optimal route for surveying the objects of interest during the mission, but most of them are designed only for stationary objects, which greatly limits their applicability for solving a number of urgent tasks. Considering existing algorithms for stationary objects, the best route produced may be far from the optimal one if at least one of the observed objects is moving. Therefore, there is a need to develop an algorithm that takes into account the mobility of the observed objects while trying to find the optimal route of detour that will allow the optimal use of the available resources (time, fuel) and hence will increase the efficiency of monitoring. In this algorithm the targets' matrix is formed as two steps in the vector form; a set with the following values: coordinates for each object (φ_i, λ_i), angle of the course for each object (Course i), each object velocity (V_i), the RPA speed (V_0), is sent to the input of the open matrix. In the first step, the relative positions of the objects relative to the current position of the RPA are calculated. The algorithm then calculates the time t , required to approach the object, and the object is excluded from further processing. Then the new positions taken by all objects during time t are calculated. Afterwards the algorithm starts a next iteration to update the coordinates of the objects. If illegal pollution from a ship or an accidental pollution in the object area are found, according to the dedicated surveillance algorithm, this object in the surveillance field of the camera of the RPA will be checked using the RPA sensors payload. In the second step, the matrix with the calculated new positions and distances of the objects will be used as the cost matrix for the optimization algorithm of the travelling sales person.

The optimization model was proposed for the RPA route. The model considered two scenarios. In the first scenario, all target coordinates were inserted in matrix, while in the second scenario some of the objects moved with a constant speed and course. An example graph was proposed with 11 targets, assuming that there are 5 targets with a speed of 12 knots, 3 targets with a speed of 8 knots, and 3 static targets. In the second scenario, the coordinates were calculated in defined intervals of time. Scenario 1: the RPA was employed for monitoring 11 targets, with an algorithm used to find the shortest route. Scenario 2: the RPA was employed for monitoring 11 targets; the impact of time intervals, was considered, route planning was monitored, with the aim to minimize the total RPA route distance. Comparing the solution of scenario 2 with the solution in scenario 1, the optimal route distance decreased by 7%. This result demonstrated an objective value, and the route improved significantly.

To solve the problem of finding the optimized route the tree algorithm, dynamical programming, method of the branch and bound algorithm were used. For the tree algorithm a "start" vertex $S \in V[G]$ was selected, and the Prim algorithm was used to compute a minimum spanning tree from s . Then the program will assume L to be the sequence of vertices visited in a tree walk, and the Hamilton cycle that visits the vertices in the order L will be returned. The disadvantage of this algorithm is that this is an approximate method, and the whole vertex of the graph should be checked as the starting point, since a longer time is needed to perform calculations for graphs with more than five vertexes (Kanthavel 2016). For the analysis of all algorithms for quantities of vertices from 5 to 30, 15 different graphs were generated, with different cost matrixes and edge weights. For each graph, the optimal route was found. The results of the performed calculations show that the tree algorithm has a very small accuracy for the number of vertexes greater than five. This problem restricts the implementation of this algorithm for solving the surveillance mission in practice.

In the dynamic programming method, the content of the RPA flight problem algorithm in the k step, $1 \leq k \leq n-1$, requires that the existing surveillance route of the RPA "length k " (passing through the k vertices) is extended to l , for example, the path of passing through the $(k+1)$ vertices is considered. For representation of the route optimization of the RPA performing a surveillance mission, it must be taken into account that there is an optimal route $L = [0, j_1 j_2, \dots, j_n]$. In the algorithm, the leg of the route which was started in 0 and finished in vertex I , $L_{0,i} = [0, j_1 j_2, \dots, j_k, i]$ will be analyzed, this route includes k of intermediate vertices. If L is the minimal path, the part of it $L_{0,i}$ that connected vertices 0 and i and is going across vertices $j_1 j_2, \dots, j_k$ in some order should have a minimal length. The length of the shortest route that connects vertices 0 and i and passes once and only once through each k of vertices $j_1 j_2, \dots, j_k$ is $f_{oi}(j_1 j_2, \dots, j_k)$. The principle of the optimization of the RPA routing in dynamic programming is to find the minimum of function f_{oi}^k . The implementation of the dynamic programming method for the RPA routing problem is exponentially dependent on the size of the input data. For a mission with more than 30 objects, the proposed method will need more time. The results of the performed calculations show that the dynamical programming algorithm has a normal accuracy.

The branch and bound algorithm is based on the idea of sequential decomposition of a set of feasible solutions. At each step of the method, the elements of partitioning (subset) are analysed so as to include or not include a given subset optimal solution (Clausen, Traff 1991). If there is a problem to find a minimum, a check can be carried out by comparing the values of the lower

bound of the objective function in this subset sets with an upper bound function. For the RPA surveillance mission route, the problem can be formulated as $f(x) \rightarrow \min_{x \in S}$, where $f(x)$ is the real function, and S is a finite set of admissible solutions. This way a minimal route should be found. The mission plan is specified as a full directed graph $G = (V, E)$ with a plurality of vertices $V = \{1, \dots, n\}$ and a plurality of edges E . For each edge $(i, j) \in E$, length $c_{ij} \geq 0$ is assigned. A simple route from i_1 to i_k is $\{i_1, \dots, i_k\}$ and the length of the route $\{i_1, \dots, i_n, i_1\}$ is $f(i_1, \dots, i_n)$, taking into account that $i_1 = 1$. The subset of feasible solutions is defined as a pair of sets (I, J) , where $I = \{i_1, \dots, i_k\}$ is the partial route (a sequence of visits to first k nodes), $J = \{j_1, \dots, j_q\} \subset V \setminus I$ – a set of restrictions on the flight to the last point i_k of a partial route I . To represent the sets of partition in $V' = V \setminus I \cup J$ an element i is chosen. If the element i in V has only a single element p , the desired set $I = \{i_1, \dots, i_k, i\}, J = \emptyset$ and $(I = \{i_1, \dots, i_p, k\}, J = \emptyset)$. The next step is to find the lower bound $H(I, J)$ for subset (I, J) . When all elements are visible, the decomposition algorithm completes the work, and the current record is the best solution. In an opposite case, the set that is perspective will be selected from the unchecked elements. It is subject to partition (branching). The process continues until all elements of the partition are checked. This algorithm has a high accuracy and is not limited by the number of mission points to be visited. If the mission task includes many points, it is better to use the branch and bound algorithm.

3. Conclusions

To solve the route optimization problem, the task of the surveillance mission can be represented as the process of visiting all nodes of the graph $G(V, E)$. To visit a matrix of targets not limited by the number of objects, the branch and bound algorithm can be recommended for determining the optimal route. The example showed that the objective value for 2 scenarios improved significantly. It demonstrates that the proposed optimization method for RPA route planning is effective.

References

- Belmann, R. 1957. *Dynamic programming*. Princeton, NJ, USA: Princeton University Press. 342 p.
- Clausen, J.; Traff, J. 1991. Implementation of parallel Branch-and-Bound algorithms – experiences with the graph partitioning problem, *Annals of Operation Research* 33(5): 331–349. <https://doi.org/10.1007/BF02073939>
- Dijkstra, E. 1959. A note on two problems in connection with graphs, *Numerische Mathematik* 1: 269–271. <https://doi.org/10.1007/BF01386390>
- Guerin, R.; Orda, A. 2002. Computing shortest paths for any number of hops, *IEEE/ACM Transactions on Networking* 10(5): 613–620. <https://doi.org/10.1109/TNET.2002.803917>
- Hung, M.; Divoky, J. 1988. A computational study of efficient shortest path algorithms, *Computers & Operations Research* 15(6): 567–576. [https://doi.org/10.1016/0305-0548\(88\)90052-4](https://doi.org/10.1016/0305-0548(88)90052-4)
- Kanthavel, R. 2016. Developing Dijkstra-Primbert Algorithm for finding unpredictable paths over time-varying networks, *Circuits and Systems* 7: 3541–3555. <https://doi.org/10.4236/cs.2016.711301>
- Stipanović, M.; Inalhan, G.; Teo, R.; Tomlin, C. 2004. Decentralized overlapping control of a formation of unmanned aerial vehicles, *Automatica* 40: 1285–1296. <https://doi.org/10.1016/j.automatica.2004.02.017>