AIR SPACE ROUTING AND FLIGHTS PLANNING: A PROBLEM STATEMENT AND DISCUSSION OF APPROACHES TO SOLUTION

Assoc. Prof. PhD. Alieksieiev V.

Lviv Polytechnic National University, Lviv, Ukraine

Vladyslav.I.Alieksieiev@lpnu.ua, https://orcid.org/0000-0003-0712-0120

Abstract: *Air routing has become an important problem of recent years. Wide implementation of idea to use a free routing airspace (FRA) over the Europe and idea of exploiting FRA as a main airspace management resource to reduce air traffic problems revealed a necessity of a new look to a routing problem. Many previous solutions relied on predefined topology of airways and ability to exploit well-developed methods known in graph theory. Meanwhile the problem was current due to many factors needed to be involved in the airspace as a 3D-space: air management restrictions and different air spaces regulation rules, weather conditions, danger areas, aircraft's characteristics, pilots' preferences, etc. Moreover, the appearance of FRA has made it inappropriate to use previous algorithms. Most of these algorithms required a definite topology with known routing points connected with predefined edges, while the FRA may have only border points to fly into or fly out of the area and no definite edges inside. The task of constructing the route became the same difficult as obvious: any pilot can fly directly through the FRA, but the route should be built and confirmed prior to a take-off. Problem comes even more evident if considered for the unmanned flying vehicles (UFV) and the need for robots or AI systems to solve the routing problem by it-selves. As a topping of the complexity of the problem, one may consider the upcoming difficulties of airspace congestion in FRA. Despite the problem is known for areas close to airports, it is still current to plan routes avoiding flights conflicts in the air and to avoid FRA high congestion. There are different researches on some particular problems and some approaches to solve these problems. Nevertheless, there is no complex problem statement yet. This research was focused on need of understanding the full scope of problems for air routing to understand the ability to build an efficient solution for the problem as a whole.*

Keywords: ROUTING PROBLEM, AIR ROUTING, FLIGHTS MANAGEMENT, FLIGHTS OPTIMISATION, ALGORITHM ANALYSIS

1. Introduction

The problem of air routing nowadays comes with a few starting points:

- The main goal is to find route between departure and destination airports.
- The second goal typically is to have some reduce of fuel burn. Nevertheless, one should mention that there are estimates that optimal routes are able to give typically only 1–3% of fuel burn reduce.
- It is also assumed that all routes exist within the air space that can be predefined typically both by lateral dimension boundaries and altitude limits.
- It is assumed that there are many possible routes in airspace should exist and there are some routes that can be considered optimal (or at least one). Nevertheless, one can consider situation of reaching conditions that any of these routes are inappropriate for the particular type of aircraft (i.e. with respect to its fuel capacity and/or possible altitude limit).
- A longer route is expected to be closer to “straight” route (a great circle route).

Most researches on air routing problem are made with some of following assumptions:

- the most valuable optimization could be made within cruise stage (due to it is the main part of any flight with the most fuel burned, and climb/descend stages are much shorter or can not be optimized because of predefined fuel burn rates etc.);
- the problem of 3D routing is often reduced to 2D routing considering only cruise flight stage to be able to make notable influence;
- the problem of finding the shortest path is considered solvable with some graph algorithm like Dijkstra or A* (same as in on ground routing) without respect to vertical trajectories optimization and/or wind optimization (nevertheless these algorithms are less useful in FRA – Free Routing Area).

One recent research [1] showed the great decrease of algorithm running time with a wind optimized approach algorithm compared to classical shortest path search on graph (like Dijkstra and A* algorithms). Nevertheless, the algorithm does not consider so named 3D-routing. There are also some other researches dedicated to routing in FRA [2].

Some other researchers [3-4] offer considering both weather conditions (winds, temperature) and flight altitudes and they showed that it is more efficient to consider all conditions while building the path compared to step by step consideration of each type of conditions after initial shortest path search. There are also some similar direction researches [5-8].

However, most of approaches offered do not consider possible RADs (Route Availability Document) influence. This means there should be used a combined approach to be able to find a complex air route through both a FRA and a standard airspace (with a regular net of airways). One of possible solutions with respect to PRDs (Prohibited, Restricted and Dangerous areas) was offered in [9] based on China airspace experience.

2. Prerequisites and means for solving the problem

Now, the current airspace routing problem understanding includes a set of the following problems:

- *Path construction* – a problem to find a path between airports of departure and destination with respect to RAD constraints: avoid, mandatory, etc.
- *Path optimization* – a problem to have best path according to:
  - Length (shortest flight distance)
  - Time (fastest flight)
  - Fuel consumption (cheapest flight)
  - Weather conditions: wind, temperature, etc. (both fastest and cheapest flight)
- *Path compliance* – a problem to have an appropriate path including:
  - Vertical path profile (heights or flight levels distribution over the path)
  - Smoothness (pilot preferences compliant flight and smooth path) – pilot requested path options (waypoints, segments, areas, avoid areas etc.), ability to fly (matching aircraft technical abilities, avoiding sharp turns and impossible maneuvers, pilots and passengers convenient maneuvers).

Solving the problem requires a set of goals to be defined. It is obvious that a software should be developed also. The following goals in problem solution are ordered according to its importance decrease.

1) *100% success in route construction between departure and destination points*. Success rate must involve as a successful result a completely correct answer whether the path exists or not (this mean that if the route search failed then the customer should be completely convinced that there is no any possible route in given circumstances to fly).

2) *Build a set of fully optimized routes*. The set of resulting routes must include three offered routes (in some ideal circumstances this would be the same single route):

    a. the shortest route – distance optimized only,

    b. the fastest route – time & fuel optimized (evidently, with respect to weather conditions and fuel consumption),

    c. the cheapest route – cost optimized (evidently, with respect to weather conditions, fuel consumption and FIR costs).

3) *Quickest delivery time for the first found route*. Possibly this time should not exceed 30 seconds. Next, the customer must see the progress in optimization, so each next optimized route (better than initial and each previous route) should be delivered in less than each next 30 seconds. Route delivery delay for more than 2 or 3 minutes should be considered as a long search and customer should be offered to decide whether to keep searching or to use the latest found route.

4) *Separate processing of route inexistence*. This should be a response to customer's request to perform totally route availability search, which may take a longer time. The decision whether to wait a longer time should be passed to customer.

Many researches still consider routing problem in air space as a plain problem. Nevertheless, it remains a 3D problem and according to restrictions and peculiarities of an air space, there is a set of task to be solved separately:

- Take-off / Climb and Landing / Descend. According to SID/STAR configuration there should be found a set of actual initial points to start routing. These points are evidently different to a departure and destination points of airports. This yields the path search between two sets of points (ADEP_SET and ADES_SET) instead of search between only two points (ADEP and ADES). Vertical profile for take-off (including direct climb and/or step-climb) and landing (descend) should be considered at current step. Regarding a set of flight levels one may expect up to 64 points on both ADEP_SET and ADES_SET ends for the routing (i.e. 3–8 FL × 4–8 take-off/landing edges = 12–64 initial routing points).
- Determined flight (flight via airways). According to known and predefined air routes, there should be a known solution to find route on a graph (a topology). The difficulty of the problem is due to restrictions set (like RADs) and combinatorial explosion (regarding a huge or fast growing number of involved waypoints and edges).
- "Free" flight (flight via FRA). Peculiarity of the problem for the general routing problem approach is that there are no predefined edges between waypoints and it requires a topology to be generated "artificially" to be able to solve the problem with the same algorithm like in case of flight via airways. This problem actually is a problem of connecting two sets of points: area fly-in and area fly-out. Despite of seeming simplicity of a free flight, the problem looks quite different and requires some kind of particular solution to be used.
- Short flight route. This could be considered as a special case of routing problem concerning only to build the shortest route, while the problem of optimization can be abandoned. The actual result of optimization can be less useful in most circumstances of a short flight. This can be yielded with a higher cost of optimization efforts compared to implementation of a simple direct flight (shortest path flight) or insignificant benefit of optimization.

One should remember to define, at least formally, a set of metrics to be able to understand finally if the route fits all the needs. These metrics may help to make both a better route during route construction procedure and allow a pilot to decide whether he is satisfied with a route offered. Among others, a set of metrics to understand the quality of routes should include following estimates:

- Successfulness – existence of a route
- Smoothness – calculation of route direction changes
- Duration – overall required time
- Cost – overall fuel consumption + FIR costs

3. Solution of the examined problem

There is set of a very interesting and promising approaches based on a so named "nature inspired approaches". Those considered are mostly the optimization algorithms, like ant algorithm, artificial bee colony algorithm (used in [3]), blind naked mole rat algorithm [10], rolling swarm of locust model [11], and grasshopper optimization algorithm [12–13]. Nevertheless, some of these algorithms have not been applied to solve of an air space routing problem. Here in this paper I offer to exploit the idea of a locust swarm move due to its behavior similarity to aircrafts flights. First, a locust swarm acts as a space oriented unit – it moves in particular direction like having some target. Second, a locust swarm flies actually and it is vulnerable to winds – this is very similar to aircraft flights wind vulnerability. Third, the algorithm can be used both to construct and to optimize routes.

Let's consider an artificial locust swarm behavior as a model for multi-routing solution approach. Each single locust could be considered as a solution for routing problem. Each locust in a swarm of N species can act in a definite way: "jump & fly" or "land on & eat".

"Jump & fly" should be made between waypoints $p_i \in P$ with number of waypoints $|P|$. These waypoints are considered to be a food source for locust. The initial food quantity at each waypoint can be eaten by a single locust or a group of locusts, so the number $q_i = q(p_i) \in [0...N]$ of locust species able to eat at particular waypoint could be measured also as a quantity of food at the waypoint. The number $q_i = q(p_i)$ may also define a maximum number of locust species able to stay at waypoint p_i. The set of $q(p_i) \in Q \subset \mathbb{N} \cup \{0\}$ can be included in set of parameters for the model. Each waypoint can be considered to have the same quantity of food $\forall\{i,j\} \subset \mathbb{N}$, $i \leq |P|$, $j \leq |P|$: $Q(p_i) = Q(p_j)$ or different quantity of food $\exists\{i,j\} \subset \mathbb{N}$, $i \neq j$, $i \leq |P|$, $j \leq |P|$: $Q(p_i) \neq Q(p_j)$. This should depend on problem specifics – the simplest approach to routing can be considered with $\forall i \in [0...|P|]$: $q_i = 1$.

After locust eat it have to move forward. The direction for the swarm can be generally defined by a vector $\vec{D}$ from initial point to destination point. This vector also gives a line of attraction for the swarm (and each locust in a swarm, respectively). So, the swarm would have two attraction forces: swarm self-attraction force to keep swarm together and swarm direction vector attraction force.

According to eat action each locust will choose the next waypoint filled with a food. The hunger will force the locust and a swarm to move forward and not to return back to "empty" waypoints.

The jump is an act of movement from one waypoint to next waypoint. When a locust jumps to next waypoint with a food it may be considered mandatory to eat a portion of food – the quantity of food at the point to be decremented. Each locust can "decide" to stay and eat if there is still a food at the waypoint and the swarm attraction not forces it to move again. This "decision" can be implemented via act of zero-length jump (means to stay at a waypoint). All jumps should be made over the predefined topology edges (if there are airways topology) or to some neighboring waypoint (if there are no topology edges, like in FRA).

The fly action is a situation, when there are no vacant and non-empty neighboring waypoints. In case of predefined topology, some next to neighbor waypoint can be used with respect to swarm attraction force and swarm direction vector. In case of absence of predefined topology edges (FRA) the next waypoint can be selected only with respect to presence of food and attraction forces (swarm and direction). The fly distance should be regulated with a gravity force (the distance is shorter if the force is greater).

The swarm attraction force can be used also to find a way between and around obstacles. It is expected that there could be some waypoints on the border of each obstacle, and a part of a swarm should stick around an obstacle at each of such waypoints. This means the waypoints on the border of the obstacle is being saturated to prevent other locusts to stay at the waypoint. And when the swarm moves forward behind the obstacle the swarm attraction should force the locusts from behind to move and follow the swarm.

It may happen that the swarm can divide into two swarms, if the swarm attraction force for some part of locusts becomes low. But we leave this case for further consideration.

Swarm attraction force can be calculated as a vector from a single locust position to a middle point of all species locations: $\frac{1}{N}\sum_{n=1}^{N} p_n$. Alternatively, swarm attraction force can be understood as a swarm noise. Once a single locust finds a food it may be considered to keep quiet. Contrary, there can be a sound of locust flight, attracting all locust species left at the back to follow the swarm. This may affect the locusts from the back to choose direction to the swarm forward position and not to repeat some curved ways. So, the swarm attraction force should be calculated for each locust as a vector from its current position to the swarm "sound", produced with those species in-flight. If we have a subset of locusts performed a fly, then there is a flight vector for each flying/jumping locust defined with its initial and destination position and a sum vector of the swarm noise can be found. Nevertheless, this alternate approach requires more calculations instead of having only predefined waypoints position, and this may yield inappropriate computational difficulty of the algorithm.

4. Results and discussion

A custom software solution for the algorithm was developed using Microsoft Visual Studio 2015 in C++ programming language. Currently, an initial test of a locust swarm algorithm for solving routing problem was made as a plain routing with randomly generated topologies. Simulations were performed on a Dell Inspirion notebook (model no. 3737-5683) with Intel Core i7 processor. The results of simulations are presented in Table below.

Simulations were made for different topology sizes, described with number of waypoints and number of edges in a graph. Different sizes of the locust swarms were used to find out which swarm size could be enough to find best route. The number of steps is the number of waypoints between starting and finishing points. The time of each simulation is given in milliseconds. The route length is expected to be a dimensionless value according to random generated waypoints' coordinates.

As one can mention, swarm size was selected as a value compared to number of waypoints. Nevertheless, a test of a single locust ability to find a route was made. It was found, that in case of a large topology (250 and 1000 waypoints) a single locust was unable to find a route between starting and finishing point within a predefined limit of iterations. Other swarm sizes were calculated as a 1/10, 1/5, 1/4, 1/3, 1/2, 2/3, 3/4, 4/5 and 1/1 of number of waypoints.

For small topologies (30, 50 and 100 waypoints), the shortest route was found by the swarm with number of locusts in it not less than a half of number of waypoints (size of swarm ≥15, ≥25 and ≥50 respectively). However, for bigger topologies (250 and 1000 waypoints) the size of the swarm needed to find the shortest route decreased (size of swarm ≥62 and ≥100) revealing that 1/4 or even 1/10 could be enough. This is a particularly interesting result with the algorithm implementation.

Among the cases considered, with the growth of a swarm size the number of steps in the algorithm can decreased faster compared to route length. For example, when the number of waypoints was 50 and the swarm size was 25, a route with 4 steps was found, but it was not the shortest one yet.

***Table**: Simulations results.*

Waypoints	Edges	Swarm size	Steps	Time (ms)	Length
30	150	1	12	16	56.4299
		3	15	63	83.1392
		6	10	32	69.7788
		7	5	31	29.2127
		10	5	31	40.4625
		15	2	32	23.4546
		20	2	16	23.4546
		21	2	15	23.4546
		24	2	25	23.4546
		30	2	31	23.4546
50	500	1	45	94	537.526
		5	22	94	287.817
		10	10	77	136.194
		12	8	78	114.170
		16	6	78	96.4703
		25	4	93	80.5464
		32	4	78	60.4997
		36	4	134	60.4997
		40	4	127	60.4997
		50	4	144	60.4997
100	1 000	1	54	172	1322.46
		10	14	156	378.330
		20	8	230	229.159
		25	6	209	160.357
		33	6	313	195.650
		50	5	344	97.1437
		66	5	389	97.1437
		75	5	488	97.1437
		80	5	531	97.1437
		100	5	615	97.1437
250	2 500	1	—	—	—
		25	14	922	911.938
		50	8	1047	648.49
		62	6	1005	404.38
		83	6	1376	404.38
		125	6	2029	404.38
		166	6	2628	404.38
		186	6	2947	404.38
		200	6	3242	404.38
		250	6	4081	404.38
1 000	10 000	1	—	—	—
		100	6	5993	2131.7
		200	6	11981	2131.7
		250	6	15089	2131.7
		333	6	20211	2131.7
		500	6	30669	2131.7
		666	6	40544	2131.7
		750	6	45541	2131.7
		800	6	48540	2131.7
		1000	6	61203	2131.7

5. Conclusion

The problem of air space routing was discussed and some key features differing the problem from a known on-ground routing problem were defined. A set of nature inspired optimization approaches was analyzed and an approach of an artificial locust swarm was chosen as an appropriate one. The algorithm of an artificial locust swarm routing (ALSR) was offered and partly developed. First results of the simulations are very promising and expected to be enhanced and improved in further researches. Some "fine tuning" features for the algorithm would be implemented to fit all the requirements, including routing in FRA and giving an appropriate route smoothness. In addition, a weather forecasts and avoidance areas should be involved in an algorithm to satisfy real flights requirements.

6. Acknowledgement

RocketRoute Limited (London, United Kingdom) sponsored the whole research, including current problem analysis and a developed algorithm of an artificial locust swarm routing (ALSR). This research results and all rights for the objects of intellectual property will belong to RocketRoute Limited (London, United Kingdom). The publication of the paper is made by permission of RocketRoute Limited (London, United Kingdom).

7. Literature

1. Flight Planning in Free Route Airspaces / Casper Kehlet Jensen, Marco Chiarandini, and Kim S. Larsen // 17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017), 2017 – 14p.
2. Conflict Free and Efficient Flight Routes Planning In Free Route Airspace / Mariusz Krzyanowski // Prace Naukowe Politechniki Warszawskiej. Transport, 2013. – P.277-285
3. Four-Dimensional Aircraft En Route Optimization Algorithm Using the Artificial Bee Colony / Murrieta-Mendoza, Alejandro; Botez, Ruxandra Mihaela; Bunel, Audric // Journal of Aerospace Information Systems – Vol. 15 Issue 6, Jun. 2018 – P. 307-334.
4. Lateral Navigation Optimization Considering Winds and Temperatures for Fixed Altitude Cruise using the Dijsktra's Algorithm / Murrieta-Mendoza, Alejandro; Botez, Ruxandra // Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Nov. 2014 – Vol. 1, 2015. – 9 p.
5. A Practical Approach For Optimizing Aircraft Trajectories In Winds / Ng, Hok K.; Sridhar, Banavar; Grabbe, Shon // IEEE/AIAA 31st Digital Avionics Systems Conference (DASC) Location: Williamsburg, VA, Oct. 14-18, 2012. – 14 p.
6. Generating Optimal Aircraft Trajectories with respect to Weather Conditions / B. Girardet, L. Lapasset, D. Delahaye, C. Rabut, and Y. Brenier. // ISIATM, 2013. – 12 p.
7. Optimizing Aircraft Trajectories with Multiple Cruise Altitudes in the Presence of Winds / Ng, Hok K.; Sridhar, Banavar; Grabbe, Shon // Journal of Aerospace Information Systems – Vol. 11, Issue 1, Jan. 2014. – P. 35-46.
8. Aircraft Path Planning under Adverse Weather Conditions / Xie Z. and Zhong Z.W. // MATEC Web of Conferences 77, 15001 (2016), ICMMR 2016 – 4 p.
9. Air route network optimization in fragmented airspace based on cellular automata / Shijin Wang, Xi Cao, Haiyun Li, Qingyun Li, Xu Hang, Yanjun Wang // Chinese Journal of Aeronautics – No. 30 (3), 2017 – P. 1184–1195.
10. A novel meta-heuristic algorithm for numerical function optimization: Blind, naked mole-rats (BNMR) algorithm / Mohammad Taherdangkoo, Mohammad Hossein Shirzadi, and Mohammad Hadi Bagheri. // Scientific Research and Essays – Vol. 7 (41), Oct. 2012 – P. 3566-3583 – Retrieved from: https://academicjournals.org/journal/SRE/article-full-text-pdf/F02820F28755
11. A model for rolling swarms of locusts / Topaz, C., Bernoff, A., Logan, S., Toolson, W. // The European Physical Journal Special Topics – Vol. 157, 2008. – P.93–109.
12. Grasshopper Optimisation Algorithm: Theory and application / Saremi S., Mirjalil S., Lewis A. // Advances in Engineering Software – Vol. 105, Mar. 2017 – P.30-47
13. Application of Grasshopper Optimization Algorithm for Constrained and Unconstrained Test Functions / A.G. Neve, G.M. Kakandikar, and O. Kulkarni // International Journal of Swarm Intelligence and Evolutionary Computation – Vol. 6, Issue 3, Art. No. 1000165, 2017. – 7 p. – Retrieved from: https://www.omicsonline.org/peer-reviewed/papplication-of-grasshopper-optimization-algorithm-for-constrained-and-unconstrained-test-functionsp-93308.html