Project Report for CS 357

Feeder routing for air-to-air refueling operations

Under the guidance of -
Dr. Kapil Ahuja

Team Members -
Mayank Tayal - 200001043
Rahul Raut - 200001064
Harsh Wardhan - 200001026

Table of Contents

1. Introduction:

According to the International Civil Aviation Organization, by 2045, the amount of scheduled passenger travel worldwide would have quadrupled (ICAO, 2018). To reduce the environmental impact, particularly the carbon footprints left, The European Commission's "Flightpath 2050" aims to reduce CO2 by 75% per passenger kilometer through technical advancement by 2050 . Additionally, from the current price of roughly 0.05[US$]kW1 h1 to the predicted price of more than double that amount by 2050 is predicted by Schmied, Wüthrich, Zah, Friedl et al. As a result, significant effort has already been undertaken to improve aircraft operations' fuel economy.

An aircraft that is being received has a probe with a fuel nozzle., will approach from behind to perform the refueling operation. Fighter jets' agility is better suited for such a maneuver than larger, commercially operated aircraft. The author proposes a refueling operation where the tanker, also known as the feeder, approaches the receiving aircraft, known as the cruiser, from behind and below and uses a so-called boom, a rigid telescopic structure less vulnerable to turbulence (see Timmermans & La Rocca, 2014). This arrangement, which is the opposite of its military equivalent, has a number of benefits. First, the passengers experience little discomfort because the maneuvering is nearly entirely performed by the feeder. Second, rather than the reverse, the feeder is kept in the cruiser's downwash. In this manner, the cruiser can be perfectly optimized for cruise flight and the maneuver without requiring more thrust.

The REsearch on a CRuiser Enabled Air Transport Environment (RECREATE) project included additional research on the cruiser-feeder method of air transportation. With the mission requirements in mind, optimal designs for both cruiser and feeder aircraft were put out to reduce their respective fuel consumption. The feeder aircraft was made to be much smaller than current military tankers to save fuel consumption on the feeder during refuelling flights, even though the cruiser was made to carry 250 people across a distance of 5000 NM (nautical miles). During each refueling flight, the feeder is able to offload roughly three cruisers with a payload of about 40 metric tonnes of fuel mass.Refueling missions were planned to last one to three hours and take place within a 250 NM radius of a refueling base. The project determined that long-distance flights account for the majority of fuel consumption overall and pinpointed potential locations for refueling bases along the most traveled routes, coming to the conclusion that commercial air-to-air refueling is possible to implement operations in a way that is both advantageous to the economy and to the environment. An operating cost savings of 4.5–6% is expected by a simulation of the entire system.

2. Problem description:

A set R of refueling requests makes up an instance of the air-to-air refueling problem. Each request r $\in$ R has a requested fuel mass M_{req}, a origin orig(r), a destination dest(r), time θ (r),and a time (r). The refueling operation must begin at the specified time (r). The endpoints of the path that the refueling process travels along are orig(r) and dest(r). The feeders run off of base b. A feeder departs from base b travels to orig(r1), refuels at time (r1), arrives at dest(r1), travels to orig(r2), and so on till returning to base b. base b from dest. This process is done in order to fulfill a sequence r1,...,rk (rk) of requests in R . Note that while feeders are permitted to wait in between requests, each request must be fulfilled exactly at the requested time (r).

Longitude/latitude pairings serve as the coordinates, and feeders and cruisers are expected to fly the shortest routes (on great circle routes) between the coordinates. We use the metric Haversine formula to calculate the distance between two coordinates, p and p, and we represent this distance as d(p, p). Additionally, we assume that the coordinates of the base, the request sources, and the request destinations are pairwise distinct for technical reasons.

Two independent optimization targets that both affect the cost the feeder fleet causes make up the feeder side of the issue: -

1. The first goal is the reduction in the feeders' fuel consumption. Earlier simulations assigned the feeders to meet the needs of the ships using a straightforward distribution mechanism. There may be solutions with much lower fuel use if the underlying combinatorial problem is studied more thoroughly.

2. The second objective is to reduce the size of the feeder fleet, which is advantageous in itself and might also result in a reduction in the size of the feeder bases. The entire cruiser feeder system gains from a decrease in the price of the essential infrastructure. An improved feeder distribution might make it possible to scale back the fleet and the ground-level feeder infrastructure.

On the part of the cruisers participating, it is expected that there won't be any flight manoeuvres. Our problem and the famous travelling salesman problem (TSP), the dial-a-ride (DARP) problem, and the vehicle routing problem (VRP) are all closely related.

3. Approach:

We will try to convert the feeder routing problem into several other forms and then try to solve the problem using many well established algorithms as discussed below:

- A multiobjective IP is used to choose the best meeting spots for the refueling procedure.
- A column generation approach allows for the breakdown of the problem's structure in order to locate subproblems.
- A branch and bound approach is used in the Branch-and-Price framework, which uses an adjusted labeling algorithm to solve the price subproblem.

In order to solve the pricing subproblem, the branch-and-price framework uses an adjusted labeling algorithm. First-come, first-served scheduling was used in the original simulation for the feeders. To decrease the necessary number of feeders, airborne feeders have been given priority over ground-based feeders. Therefore, neither fuel efficiency nor a reduction in the number of feeders at each base have been prioritized in the feeder schedule. to use branching decision-making.

4. Theory and Implementation:

We suppose that all aeroplanes always fly at the same speed v. Feeders begin their routes by ascending to the necessary altitude and end them with a fall before landing, whereas cruisers retain a stable altitude and direction. They are indistinguishable in terms of their physical traits and flight characteristics, fly in enormous circles, turn quickly between flight legs, and have similar flight patterns.

Each feeder has an Operational Empty Weight Mac and a Maximum Take-off Weight $M^{takeoff} > M^{ac}$. The difference $M^{max} := M^{takeoff} - M^{ac}$ is available to store fuel, which is either delivered to cruisers or burned during the flight of the feeder itself. Consequently, the fuel mass M(θ).

Now, we will be examining the different phases of the flight of a feeder and deriving their equations with respect to their fuel consumptions defined as follows :

- **Free flight:** During the free flight phase, the feeder maintains stable speed and altitude. Consequently, the behavior of the mechanical system representing the feeder can be analyzed using basic principles of mechanics. For a given fuel mass M^o at an initial time θ_o, the solution is given by

$$M(\theta) = \left(M^0 + M^{ac}\right) \cdot \exp\left(\frac{-(\theta - \theta_0)v}{X}\right) - M^{ac},$$

  where X := v · (L/D)/sfc denotes the aircraft efficiency

- **Climb:** A feeder must ascend from its base to the cruising altitude of the cruisers that need to be refuelled. We consider the ascent to be carried out at a constant rising angle. Feeders can start fulfilling requests only when the required amount of time has passed from takeoff.

- **Descent:** The final drop to the base can be carried out in gliding flight, but otherwise, the descent phase is essentially identical to the free flight period itself. This gliding distance is indicated by the letter $d^{gliding}$. Fuel is burned continuously during the gliding phase at a rate of $\rho^{gliding}$, regardless of the mass of the feeder. The full distance is traversed gliding if the distance from the feeder's current position to the base does not exceed $d^{gliding}$. Otherwise, a free flight phase is conducted to get close enough to $d^{gliding}$ to the base.

- **Refueling:** The refueling operation of a cruiser $r \in R$ is conducted in three steps, each of which has a constant duration. First, the feeder approaches the cruiser and connects its refueling boom to the cruisers fuel receptacle. We let $\theta^{approach}$ be the corresponding approach duration. As soon as the contact is established, fuel is pumped from feeder to cruiser at a constant rate of $Mreq(r)/\theta^{contact}$ over the wet contact duration θ contact. After the fuel transfer is completed, the boom is disconnected and the feeder retreats from the cruiser, requiring a retreat duration $\theta^{retreat}$. The approach towards and retreat from the cruiser correspond to free flights (where the difference in fuel mass is given by

$$M(\theta) = \left(M^0 + M^{ac}\right) \cdot \exp\left(\frac{-(\theta-\theta_0)v}{X}\right) - M^{ac},$$

while the change in fuel mass during the wet contact time is given by

$$\dot{M}(\theta) = -\left(\frac{M^{ac} + M(\theta)}{L/D} \cdot sfc + \frac{M^{req}(r)}{\Delta\theta^{contact}}\right).$$

The solution of this ODE for a fixed M^o is given by

$$M(\theta) = \left(M^0 + M^{equiv}(r)\right) \cdot \exp\left(\frac{-v \cdot \Delta\theta^{contact}}{X}\right) - M^{equiv}(r),$$

where $M^{equiv}(r)$ is an equivalent aircraft mass of

$$M^{equiv}(r) := M^{ac} + X\frac{M^{req}(r)}{v \cdot \Delta\theta^{contact}}.$$

The entire refueling operation has a duration of θrefuel defined as

$$\Delta\theta^{\text{refuel}} := \Delta\theta^{\text{approach}} + \Delta\theta^{\text{contact}} + \Delta\theta^{\text{retreat}}.$$

- **Base refueling:** The refueling of the feeder itself is conducted at the base before the feeder takes off to subsequently serve a number of cruisers.

5. Problem Formulation:

We define an initial fuel mass function $\boldsymbol{\mu} : \mathbf{R}^+ \times \mathbf{R}^+ \to \mathbf{R}^+$, describing the initial fuel mass depending on the final fuel mass M_{final} and the duration θ of an operational phase. In a similar fashion, we let $\boldsymbol{\Delta\mu} : \mathbf{R}^+ \times \mathbf{R}^+ \to \mathbf{R}^+$ be the fuel burn function, i.e., the fuel that is consumed by the feeders themselves.

So $\boldsymbol{\mu}$ gives us the initial fuel in the feeder if we know the final fuel in feeder and time of flight. The $\boldsymbol{\mu}$ depends on M_{final} as fuel consumption depends on the weight of the feeder which in turn depends on fuel in it.

Let Refueling Graph D = (V, A) be a acyclic graph with V as set of vertices consisting of V_{base} (set of refueling bases), V_{head} (set of request heads), V_{tail} (set of request tails) and A as set of arcs (edges) joining this vertices.

Times associated with various activities are defined as:

$$\begin{aligned}
\theta^{\text{takeoff}}(r) &:= \theta(r) - \max\left(\Delta\theta^{\text{climb}}, \frac{d(b, orig(r))}{v}\right), \\
\theta^{\text{landing}}(r) &:= \theta(r) + \Delta\theta^{\text{refuel}} + \frac{d(dest(r), b)}{v}, \text{ and} \\
\theta^{\text{base,refuel}}(r) &:= \theta^{\text{takeoff}}(r) - \Delta\theta^{\text{base,refuel}}.
\end{aligned}$$

Where $\Delta\theta^{\text{base,refuel}}$ is the time associated with refueling of feeder at base, which we have assumed to be constant.

Further we divide the set of Arcs A into various subsets based on the activity our feeder is doing along, they are A^{req}, A^{climb}, $A^{descent}$, A^{flight}, $A^{base,refuel}$, and A^{wait} (idle flight to wait for incoming cruiser). Out of them we form a subset $A^{transit}$ as $A^{transit} = A^{climb} \cup A^{descent} \cup A^{flight}$.

Now our aim is to cover all the request arcs (A^{req}) in the above refueling graph consuming the minimum amount of fuels by the feeders and also to minimize the number of feeders used.

Thus our optimization problem is the minimization of $\Delta\boldsymbol{\mu}$ (the fuel burn function) subject to the constraint that every request $r \in R$ must be satisfied.

Let $P : (a_1, a_2, \ldots, a_{k-1})$ be a path with arcs $a_i : (u_i, u_{i+1})$ i.e. an edge from vertex u_i to vertex u_{i+1}. Let V_p be set of vertices covered by P, $V(P) : \{u_i \mid i= 1, \ldots, k\}$.
The initial fuel mass at start of the path can be recursively defined as:

$$\mu_P(u_i) := \begin{cases} 0 & \text{if } i = k, \\ \mu(a_i, \mu_P(u_{i+1})) & \text{if } i < k. \end{cases}$$

We call a path **feasible** if and only if $\boldsymbol{\mu_P(u_i) \leq M^{max}}$ (M^{max} is the maximum fuel the feeder can carry) for all $u_i \in V(P)$. We denote the set of all feasible paths as $\mathcal{P}$. There is a one-to-one correspondence between the trajectories of the feeders and feasible paths.

Now our problem can be modeled as a set covering formulation with our universal set being the set of all request arcs i.e. $\mathbf{A^{req}}$. We have the set of feasible paths $\mathcal{P}$, out of which we should pick up a combination of feasible paths such that we cover each and every request arc at least once.

Our optimization problem thus is,

$$\min \sum_{P \in \mathcal{P}} c_P x_P$$

$$\text{such that,} \quad \sum_{P \in \mathcal{P}:\, a \in P} x_P \geq 1 \qquad \forall\, a \in A^{req}$$

$$\sum_{P \in \mathcal{P}:\, a \in P} x_P = y_a \quad \forall\ a \in A^{\text{transit}}$$

$$y_a \in \{0, 1\} \qquad \forall\ a \in A^{\text{transit}}$$

$$x_P \in \{0, 1\} \qquad \forall\ P \in \mathcal{P}$$

where x_P denotes whether the feasible path P is included in our optimal solution set or not. c_P is the cost associated with the path P, thus if P taken $x_p = 1$ and it adds c_P towards our cost. All the transit arcs in the graph can be covered in at most one of feasible paths in the optimal solution. y_a are the linking variables used to ensure above constraint.

6. Challenges:

The leading challenge in solving the above optimization problem is the potentially large (exponential in terms of requests) number of feasible paths and thus the x variables. Also the approach to generate all the feasible paths and then solve the resulting problem is unsatisfactory, as we will not include all the feasible paths in our final optimal solution and most of them will be unnecessary to check. Instead we decide to generate the feasible paths on-demand as and when required.

To overcome the above challenges, we employ the branch and price framework. We use the branch and bound algorithm which is nothing but modified backtracking with bounding the solution by linear relaxation and thus avoiding to expand non-optimal subtrees. That is at each node we check the best solution we can get from here-on, then compare it with the best solution we have till now, if the best value is better than the best solution obtained we proceed else we discard the entire subtree. At each node we employ a column generation technique to produce a feasible path with reduced cost, thus no need of knowing all the feasible paths at the start of the problem.

7. Algorithm:

We run a branch and bound algorithm at the top-level. We first solve the relaxed Linear Program of the original to get a lower bound on our cost. After that, at each node take branching decisions to include further paths. After each decision we calculate the minimum attainable cost (optimistic cost) from that node, if that is worse than the best solution we have already reached at some prior point we discard the node and don’t expand it further, else we continue with normal expansion. Pseudo-Code is given below:

PseudoCode:

```
begin
    nodes_to_explore :={∅};
    minimum_cost:=solve relaxed LP;
    paths_included:=NULL;
    while nodes_to_explore is not empty do
        choose a branching node, node k ∈ nodes_to_explore;
        remove node k from nodes_to_explore;
        // generate paths (p1, p2, ..., pk) which reduce cost using column
generation and
        // calculate the best possible cost you can have after branching
be    best_now
        // For this execute the pricing loop algorithm on this node
        pricing_loop()
        for i=1 to pk do
            if best_now worse than minimum_cost then
                kill child i;
            else if child is a complete solution then
                minimum_cost:=best_now, paths_included:=set of nodes from
root to child;
            else
                add child i to nodes_to_explore
        end for
    end while
end
```

We execute the pricing loop algorithm at each node of the above Branch and bound algorithm:
The pricing problem is to find a feasible path p with negative reduced cost.

The reduced cost is given by:

$$\bar{c}_P := c_P - \sum_{a \in P \cap A^{\mathrm{req}}} \lambda_a - \sum_{a \in P \cap A^{\mathrm{transit}}} \delta_a.$$

In the below code we find a path with $c_p < 0$.

Algorithm 1 :

```
function pricing_loop()
    while True do
        Solve the node LP to get x*, y*, and duals lamda*, delta*
        Pout = labeling_scheme();
        //labeling scheme given below

        if (c_Pout<0)
            nodes_graph[LP].second += Pout;
            continue;

        for( each r in the set R such that a := (u,v) )

            if ( there exists 'node' belonging to neighborhood of u
                and y[node] does not belongs to {0,1})
                A_fract = node;

            if ( there exists 'node' belonging to neighborhood of u
                and node contains compound variable y[a])
                A_comp = node;

            if ( there exists a 'node' in the A_fract such that
                intersection of A_comp and A_fract for that 'node'
                is null space then):
                compound_variables[LP].push_back(y[index]);
                Continue;

        break;
```

Labeling scheme is a sub-function called by the iteration of pricing loop function to check the feasibilty of the iteration i.e. branch and bound path. It is used to find the shortest path length under the given constraints. It requires the fuel and cost function for its computation. It takes input as directed acyclic graph, topological ordering of the nodes of the graph, cost and fuel bounds and cost and fuel functions to be able to calculate the initial requirement and taking all these we get the minimum cost requirement as the output in terms of fuel consumption.

The labeling scheme to find feasible paths and their cost is given below:

Algorithm 2 :

```
function labeling_scheme(D, s, c, mu)

    L[u] <- NULL for all u belonging to V except t
    L[t] <- {0,0}
    Cout <- Infinity
    Pout <- NULL

    for j <- n downto 1 :

        for each l_j belonging to L[u_j] :

            l_j <- (M_j, c_j)
            P_j <- corresponding_path to l_j

           for each a belonging to neighbouhood of u_j :
                a <- (u_i, u_j)
                M_i <- mu(a, M_j)
                c_i <- c_j + c(a, M_j);

                if (c_i + _c(s,u_i)) >= Cout :
                    continue

                if (M_i + _mu(s,u_i)) >= Mmax :
                    continue
                else :
                    P <- corresponding_path( _c(s, u_i))
                    if P is feasible and c_P < Cout :
                        Cout <- c_P
                        Pout <- P

                if (M_i, c_i) is not dominated in L[u_i] :
                    remove last inserted value from L[u_i]
                    insert (M_i, c_i) into L[u_i]

    for each l_s belonging to L[s] :
        l_s <- (M,c)
        P <- corresponding_path to L[s]

        if c_P < Cout :
            (Cout, Pout) <- (c_P, P)

    return Pout;
```

8. Conclusion:

We tried to decrease the fuel consumption of the passenger cruisers using the innovative idea of in-flight air-to-air refueling through light-weight feeders. We studied the physical aspects and mathematically modeled the system in the form of a refueling graph with edges as flight arcs and vertices as request points and base stations. Using the concept of feasible paths, we then formulated our problem of minimizing the fuel consumption in the form of a set covering problem. We solved it using the branch and price framework.

Our solution more than halved the fuel consumption of the air traffic on several instances. The idea has practical possibilities and can be expected to be implemented in as near as 20 years from now. The formulation can be further improved as our current formulation allows two feasible paths in our optimal solution to contain the same refueling arc. This can be avoided. Also we can try out some other algorithms to solve the obtained problem.

Lastly, in our model we had strict refuel request times and points. We need to think of more flexible and practical solutions, which may allow a time window and point range for refueling operations.

9. References and Citations:

- European Journal of Operational Research, Volume 304, Issue 2, 16 January 2023, Feeder routing for air-to-air refueling operations

- Column Generation, MTech Seminar Report by Soumitra Pal

- Column Generation - branch and price, cutting stock, Discrete Optimization, Constraint Programming, Coursera

- Branch and Bound Algorithm, geeksforgeeks