A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem

Varun

December 6, 2024

---

1 INTRODUCTION

The hub-and-spoke network model, commonly used by commercial airlines, is efficient but highly susceptible to disruptions such as flight delays, reduced airport capacity, and weather or crew-related challenges. These disruptions can escalate operational costs significantly—for instance, U.S. air transportation delays in 2007 amounted to $32.9 billion (Ball et al. **?**). Recovery planning often adopts a sequential approach for aircraft, crew, and passengers, leading to suboptimal outcomes and neglecting passenger satisfaction. Furthermore, existing recovery methods, including network flow models, integer linear programming, and ant colony optimization, face limitations such as computational inefficiency, lack of scalability, and an inability to adapt to real-world constraints.

This paper addresses these challenges by proposing a column generation post-optimization heuristic that integrates aircraft and passenger recovery into a unified framework. The proposed method ensures computational efficiency while considering operational constraints like airport capacities and maintenance schedules. It first fixes the aircraft rotations by using LNS and then assigns passengers to trips using column-generation heuristics.

2 PROBLEM FORMULATION

Instead of solving it jointly we formulateda sequential model which 1st optimises the rotations of aircrafts and then assigns passengers to flights.

Notations can be seen below in table

Table 1 – *Problem Notation*

Symbol	Description
Sets	
N	Set of airport nodes
$A = \{(i,j)\}$	Set of arcs representing flight legs or connections between airport nodes
F	Set of aircraft, indexed by f
K	Set of passenger itineraries, indexed by k
M	Set of cabin classes (economy, business, first class), indexed by m
A_f	Set of flights assigned to aircraft f in the current plan after delays
E_f	Set of feasible flight-to-flight transitions for aircraft f
r_a	Revenue (\$) from operating flight a
(o_a, d_a)	Origin and destination airports for flight a
$(C_{arr,i,t}, C_{dep,i,t})$	Maximum number of arrivals and departures at airport i within time interval t
D_k	is the set of arcs (i,j) where i is the departure node of itinerary k
Variables	
$x_{a,f} \in \{0,1\}$	1 if aircraft f operates flight a, 0 otherwise
$x_{a,a',f} \in \{0,1\}$	1 if aircraft f operates flight a' just after flight a, 0 otherwise
$y_{i,j,m,k} \geq 0$	Number of passengers in cabin class m for itinerary k on arc i to j
π^{jk}	Dual variables for flow balance constraints
λ^k	Dual variables for demand satisfaction constraints
μ^{ijm}	Dual variables for capacity constraints
Passenger-related Costs	
$c^{del}_{i,j,m,k}$	Cost of delay for passengers in cabin class m on itinerary k on arc i to j
c^{can}_k	Cost of cancelling itinerary k
$c^{down}_{k,m}$	Cost of downgrading passengers from itinerary k in cabin class m
Aircraft-related Costs	
$c^{op}_{a,f}$	Operating cost for aircraft f on flight a

Below is a simplified version of the modifying aircraft rotations as we have not considered aircraft maintenance in this formulation.

Objective:

$$\text{Minimize} \quad \sum_{a \in A} \sum_{f \in F_a} (1 - x_{af}) \cdot r_a + \sum_{(i,j) \in A} \sum_{f \in F} x_{af} \cdot c^{op}_{af}$$

Constraints:

1. **First flight constraint** (where the flight leaves its starting airport):

$$\sum_{a' \in \delta^+(source_f)} x_{f,source_f,a'} = 1$$

2. **Last flight constraint** (where the flight arrives at its final airport):

$$\sum_{a' \in \delta^-(sink_f)} x_{f,a',sink_f} = 1$$

3. **Continuity constraint for intermediary flights** (ensuring the continuity of the flight path):

$$\sum_{a' \in \delta^+(a)} x_{f,a,a'} = \sum_{a' \in \delta^-(i)} x_{f,a',a}$$

This constraint ensures that an aircraft f can only operate a flight a if flight a is part of the assigned route for aircraft f.

4. **Flight operation constraint** (ensuring that a flight is operated by an aircraft only if it is in the route of the aircraft):

$$x_{a,f} \leq \sum_{f' \in \delta^+(a)} y_{f,a',a}$$

5. **Airport capacity constraints for arrivals**:

$$\sum_{\text{aircraft } f} \sum_{\text{flight } a \text{ arriving at } i \text{ in } t} x_{a,f} \leq C_{i,t}^{arr} \cdot \alpha \quad \forall \text{ airport } i,\ t,$$

6. **Airport capacity constraints for departures**:

$$\sum_{\text{aircraft } f} \sum_{\text{flight } a \text{ departing from airport } i \text{ in } t} x_{a,f} \leq C_{i,t}^{dep} \cdot \alpha \quad \forall \text{ airport } i,\ t$$

Binary Constraints:

$$x_{a,f} \in \{0,1\} \quad \forall\ a \in A,\ f \in F_a$$

$$x_{a,a',f} \in \{0,1\} \quad \forall\ a, a' \in A,\ f \in F_a$$

This part gives us flights which are still being operated after disruptions.

2.1 Passenger Assignment Problem

The passenger assignment problem aims to allocate passengers to available flights while minimizing the total costs associated with downgrading, delay, and cancellations, all subject to various constraints like flow balance, demand satisfaction, and capacity limits. This section formulates the problem mathematically and describes the associated objective function, constraints, and dual variables involved in the optimization process. In pricing problem we calculated shortest path for each itinerary instead of adding directly variables with negative reduced cost. For example if there are 100 itinerary then we calculated 100 shortest paths and added them to master problem in each iteration. Stopping critieria we kept if the number of iterations reaches 10 or 2 minutes. CSPY was used to calculate shortest path problem.

2.2 Objective Function

The objective of the passenger assignment problem is to minimize the total costs associated with downgrading, delays, and cancellations of flight assignments. The total cost is represented by the following expression:

$$Minimize \quad Z = \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{mk}^{down} y_{ijmk} + \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{ijmk}^{del} y_{ijmk} + \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{k}^{can} y_{ijmk}$$

2.3 Constraints

The problem is subject to the following set of constraints, which ensure that passengers are assigned to flights in a feasible and optimal manner.

- **Flow Balance**: The number of passengers entering and exiting each airport must balance.

$$\sum_{(i,j)\in D_k, m\in M} y_{ijmk} - \sum_{(j,i)\in D_k, m\in M} y_{ijmk} = 0 \quad \forall k \in K, j \in N$$

  where D_k is the set of flight segments associated with demand k, and N is the set of airports.

- **Demand Satisfaction**: The total number of passengers assigned to each demand must meet the required demand.

$$\sum_{(i,j)\in D_k,m} y_{ijmk} = d_k \quad \forall k \in K$$

where d_k represents the demand for flight k.

- **Capacity Constraints**: The total number of passengers assigned to a flight cannot exceed its capacity.

$$\sum_k y_{ijmk} \leq \sum_{f\in F} cap_{jm} x_{ijf} \quad \forall (i,j) \in A, m \in M$$

where cap_{jm} is the capacity of flight segment (i,j) in mode m, and F is the set of available flights.

- **Non-Negativity**: The number of passengers assigned to each flight must be non-negative.

$$y_{ijmk} \geq 0 \quad \forall (i,j) \in A, m \in M, k \in K$$

2.4 Reduced Cost Calculation

The reduced cost is given by the following formula:

$$\bar{c}_{ijmk} = c_{ijmk} - (\pi^{jk} - \pi^{ik} + \lambda^k d_{ijk} + \mu^{ijm})$$

Where:

- c_{ijmk} represents the original cost of assigning a passenger to the flight segment.
- π^{jk}, λ^k, and μ^{ijm} are the dual variables associated with the flow, demand satisfaction, and capacity constraints, respectively.
- d_{ijk} is a binary parameter that equals 1 if the arc (i,j) is part of the flight demand D_k, and 0 otherwise.

Reduced cost was used to calculate shortest path between source to sink. As all the constraints are satisfied in RMP problem so no demand restriction was there in pricing problem.

3 EXAMPLES

In the analysis of the aircraft's rotation and the determination of compatible arcs after a disruption, we use the following condition to define compatible arcs:

$$ArrivalofFlight1 \leq TurnaroundTime + DepartureofFlight2$$

Additionally, the arrival airport of Flight 1 must be the same as the departure airport of Flight 2. These conditions ensure that the aircraft is able to complete its rotation and remain within the operational constraints for turnaround times.

Figure 1: Aircraft 3184 Rotation

Figure 1 shows the actual rotation of Aircraft A3184, with the flights numbered as follows:

3064 3069 3074 3077 3082 3085 3092 3095

These flights represent the scheduled routes of the aircraft across various airports on the given date.

Figure 2: Compatible Arcs for Aircraft 3184

Figure 2 illustrates the compatible arcs for Aircraft A3184, which are determined based on the condition described above. The compatible arcs are the pairs of flights that meet the criteria of arrival time, turnaround time, and matching departure and arrival airports.

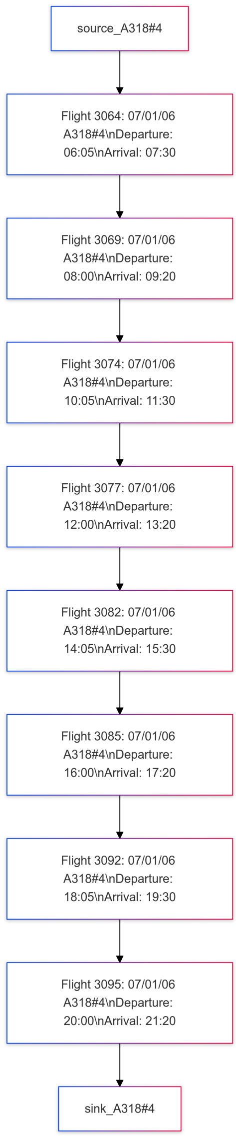

Figure 1 – *Aircraft 3184 Actual Rotation*

As seen in Figure 1, the aircraft follows a set rotation between different airports. The compatible arcs, shown in Figure 4, represent the pairs of flights that are feasible based on the turnaround and airport compatibility conditions.

Figure 3: Passenger 1011 Original Itinerary

Figure 1 shows the original itinerary for Passenger 1011, which includes the flights numbered as follows:

$$source \quad 2385 \quad 4390 \quad sink$$

These flights represent the passenger's journey starting from the source airport, with multiple stopovers at different airports, and finally ending at the destination.

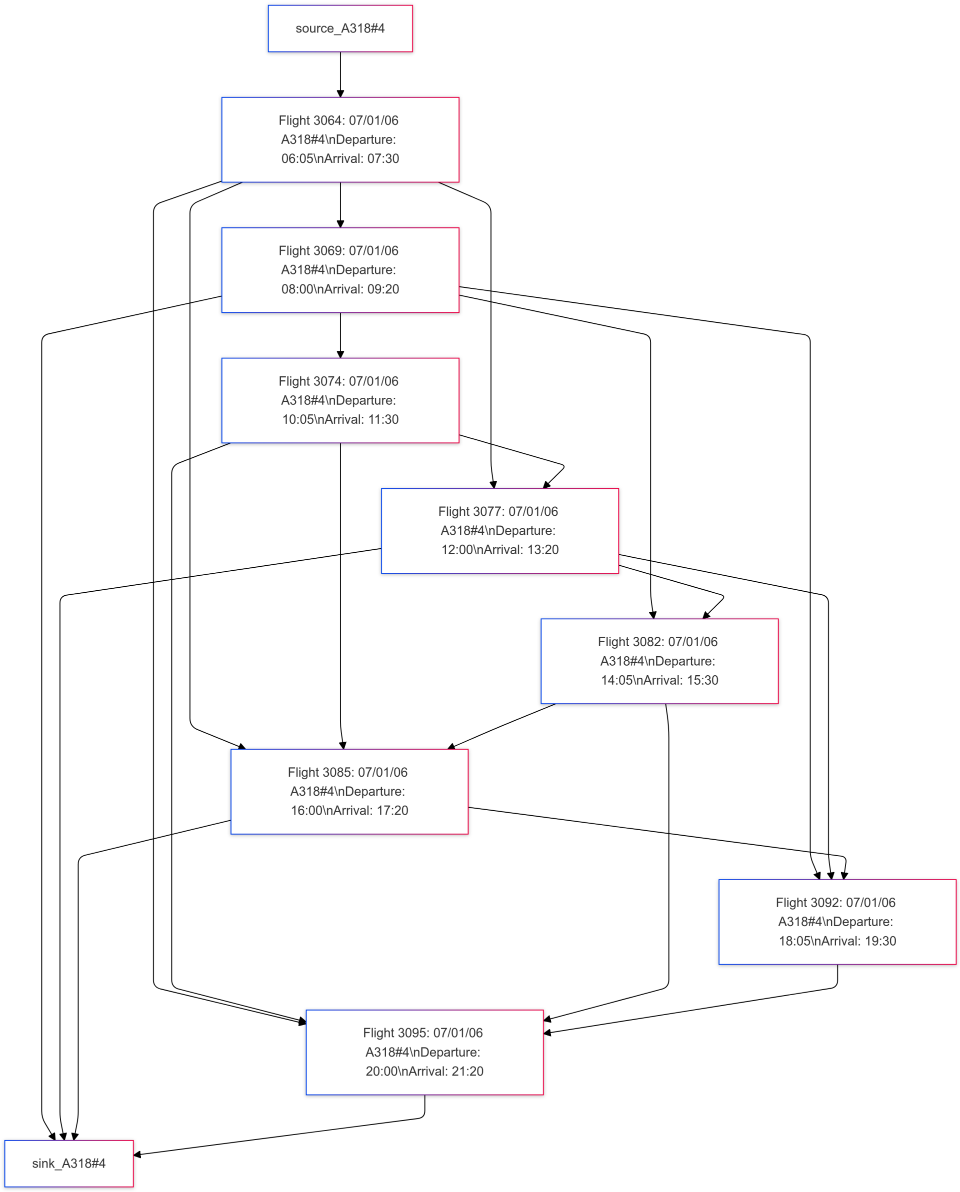

Figure 2 – *Compatible Arcs for Aircraft 3184*

Figure 4: Compatible Arcs for Passenger 1011

Given an itinerary, the first step is to check all flights departing after the first flight of the itinerary to ensure that the departure times and airports are compatible. Following this, the start and end airports are checked, verifying if flights from the start airport lead to the destination airport. Figure 3 illustrates the original flight itinerary graph, depicting various flights and their connections. Figure 4 shows an example itinerary where flight `1011` is at the core, and connecting flights are identified based on compatibility. Finally, These arcs are created by considering all flights that depart after the first flight in the itinerary, followed by flights that start at the source airport and end at the destination airport, ensuring both time and airport compatibility for each connection.

In Figure 4, the arcs represent the possible connections after ensuring that the arrival and departure times, as well as airports, match the criteria for compatibility.

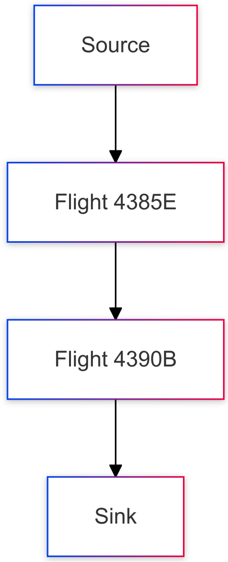

Figure 3 – *Passenger 1011 Original Itinerary*

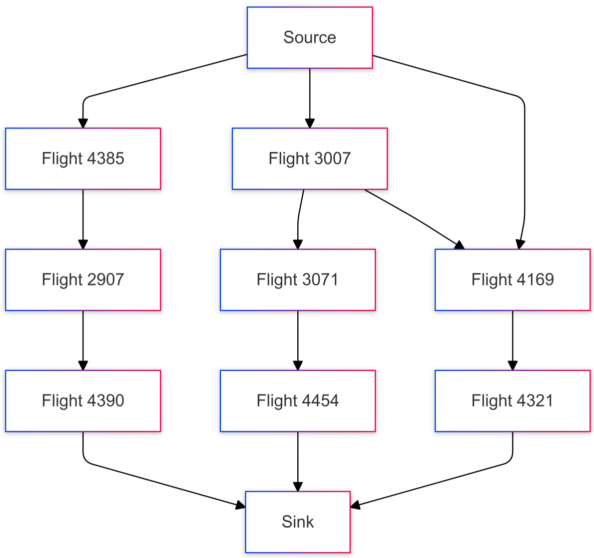

Figure 4 – *Compatible Arcs for Passenger 1011*

4 DEMONSTRATION

In our experiments we tested only on one instance with 85 aircrafts, 35 airports, 608 flights and 1943 itineraries, with 83 flight disruptions, 3 airport disruptions. The total operating costs came out to be 4.21 million euros and and delay,downgrade,cancellation cost came out to be 46.8 million dollars which is pretty high as compared to paper they have 1 million overall cost for this instance. This is because we are solving it sequentially and our input to column generation is far from optimal and number of flights are also reduced to half as can be seen from results below.
Optimal number of flights served: 291
Optimal number of passengers transported: 28377
Optimal number of aircrafts utilized: 78

4.1 EXTENSIONS

One extension could be add the crew members to the probelm. 2nd could be instead of solving sequentially we can try out integrated version of this and last instead of using LNS we can use ML methods to find intial feasible soultions.