Cornell Tech

ORIE 5380

Optimization Methods

Express Air's Cargo Operations

Author:
Maximilian Kuester

Professor:
Huseyin Topaloglu

December 6, 2018

Contents

1 Executive Summary

1.1 Problem Presentation

Express Air seeks an optimal weekly movement cycle for their 1200 planes strong fleet between three airports. Holding a cargo load in an airport results in a cost dependent on the length of the stay. Moving empty planes between the airports to satisfy the demand of the system also causes a cost. Express Air wants to minimize the total cost of their logistic operation. It is known in advance how much cargo needs to be delivered from each origin to destination airport on a particular day.

1.2 Results

With the given conditions a model has been developed that suggests an economically optimal assignment of aircrafts. Table 1 shows the optimal amount of aircrafts that need to be assigned on each route. Red numbers represent the amount of loaded aircrafts traveling from origin to destination airports, the green number represents the amount of empty aircrafts. The last three lines show how many aircrafts wait at which airport at the respective airport A, B or C.

Routes	Monday	Tuesday	Wednesday	Thursday	Friday
A → B	220 0	270 0	100 0	400 0	110 0
A → C	30 0	70 0	50 0	50 0	50 0
B → A	25 275	25 160	25 405	25 0	25 185
B → C	25 185	25 410	25 0	25 365	25 365
C → A	40 0	40 0	40 0	40 0	40 0
C → B	400 0	200 0	300 0	200 0	400 0
A → A	0	0	75	95	0
B → B	0	0	15	0	0
C → C	0	0	165	0	0

Table 1: Optimal assignment of aircrafts for each day on each leg

This solution was found applying an integer problem linear programming method. This model predicts the optimal solution under the assumption that the cost of holding one unit of cargo is 10 for each day, the cost of moving an empty plane from A to B or B to A is 7, the cost of moving an empty plane from B to C or from C to B is 6 and the cost of moving a plane from A to C or from C to is 3. under these conditions the objective value, hence the minimal expense for operating this system is **17925**.

1.3 Recommendations

The best way of cutting costs is to increase the amount of cargo that has to be shipped out from B to A and C. Ideally, to avoid any planes flying empty (85% of costs) the demand should be increased the following way:

- A-B Monday +275.0
- B-C Monday +185.0
- B-A Tuesday +160.0
- B-C Tuesday +410.0
- B-A Wednesday +405.0
- B-C Thursday +365.0

- B-A Friday +185.0
- B-C Friday +365.0

To bring costs down to 0, the fleet size needs to be additionally grown to 1390 units from 1200.

2 Problem Overview

Express Air has 1200 planes at disposal to deliver cargo between three airports A, B, and C. The company wants to know how to schedule flights between these three airport to satisfy the demand of cargo deliveries they have. Express Air knows in advance how much cargo they will have to transport between two given airports on each given day of the week. Table 3 in the Data Description section shows how much cargo comes into the system at each day. The travel time between each airport is considered one period. the distances between the respective airports are such that the travel time between each airport is of one day, implying that each period is one day. Cargo that is coming in at a certain airport can both be delivered to the destined airport or can be stored at the airport and be delivered at a later time. Storing a cargo unit is considered an expense. Also, the company can decide to re-position its planes to serve another airport on the following day. Flying an empty aircraft to another airport intuitively causes a loss. Summing up, Express Air has three decision variables for choosing how to run their daily operations:

- Amount of loaded planes that can be flown from an origin to a destination airport (PROFIT)
- Amount of empty planes that can be flown from an origin to a destination airport (including aircrafts waiting at an airport to depart on a later day)(COST)
- Cargo can be stored in an airport for a certain time period (COST)

The amount of cargo that can be moved from one airport to another depends on the amount of planes available at that location on that given day. This amount is determined by the operations of the prior day.

3 Data Description

Express Air has a fleet of 1200 planes. Each plane can transport one load unit of cargo. The fleet operates between airports A, B and C. The distance between all airports is such that the travel time for a plane from one airport to the other is one day. If a plane flies loaded there is no loss. If, however, a plane is flown empty between two two airports to be re-positioned, then there is a connected cost to it. Table 2 displays the cost of travel for empty planes between all airports. Another cost is caused by holding cargo

	A-B or B-A	B-C or C-B	C-A or A-C
Cost	7	6	3

Table 2: Cost of transferring empty planes between airports

in an airport. Keeping cargo on the ground costs 10 for each day. As anticipated, each day a defined amount of cargo is added to the system and needs to be moved according to Table 3. Therefore, if a load that has to be transported from A to B on Monday is chosen to be delivered on Tuesday instead, there is cost of 10 for each unit stored for each day. The amount of cargo that on the next day needs to be delivered from A to B is increased by that unit, too. It is assumed that the cost is 10 for keeping cargo over the weekend, considering business days as the dictating measure. Within the weekly cycle a cargo unit is allowed to stay on hold for not more than 4 business days, otherwise the system will start accumulating a potentially infinite amount of cargo at that airport when the cycle is implemented repeatedly.

Routes	Monday	Tuesday	Wednesday	Thursday	Friday
A $\rightarrow$ B	100	200	100	400	300
A $\rightarrow$ C	50	50	50	50	50
B $\rightarrow$ A	25	25	25	25	25
B $\rightarrow$ C	25	25	25	25	25
C $\rightarrow$ A	40	40	40	40	40
C $\rightarrow$ B	400	200	300	200	400

Table 3: Loads of cargo that come into the system day by day with delivery specifics

4 Overview of the Optimization Model

An optimization model is developed for Express Air. The model can be represented as a multi-state network. Each node represents an airport at a certain day of the cycle. In Figure 1 orange arrows represent full planes, green arrows empty planes and the black arrows give insight on the holding of cargo at a particular airport between two days. The model minimizes the expense the company incurs for a weekly plane movement cycle, still guaranteeing that all cargo is delivered from origin to destination.

4.1 Decision Variables

Three sets of decision variables are defined:

- amount of loaded planes flying between airports each day
- amount of empty planes flying between airports each day to re-position
- amount of cargo units hold at each airport

4.2 Data & Parameters

The data containing the amounts of cargo that is delivered on day t to each airport i to go to j is stored in $cargo_{ijt}$. Additionally, there are five main parameters:

- COST_AB = 7, is the cost of moving empty planes between A and B or vice versa
- COST_BC = 6, is the cost of moving empty planes between B and C or vice versa
- COST_AC = 3, is the cost of moving empty planes between A and C or vice versa
- COST_HOLD = 10, is the cost of holding a cargo unit for a period of one day in any of the airports
- FLEET = 1200, is the size of the fleet

4.3 Objective Function

The objective function captures the actions that cause a cost. Hence it minimizes the sum of the costs incurred by

- the amount of empty planes moved between airports
- the amount of cargo units hold in any airport

4.4 Constraints

Technically the model has 7 constraints. More details about that later. In general, however, constraints are imposed to

- ensure there are 1200 planes in the system at any given time
- ensure there are enough planes available at each airport on each day to carry the desired loads
- ensure all loads are correctly moved in the system
- ensure all cargo units entering the system are delivered within a week

5 Mathematical Details of the Optimization Model

5.1 Decision Variables

- f_{ijt} = the amount of loaded planes flying from airport i to airport j on day t, $f_{ijt} \in Z_+$
- e_{ijt} = the amount of empty planes flying from airport i to airport j on day t to re-position $e_{ijt} \in Z_+$
- h_{ijt} = the amount of cargo units hold at airport i on day t with destination at airport j $h_{ijt} \in Z_+$

5.2 Objective Function

$$\min \sum_{t=0}^{4} \Big[7 \cdot (e_{01t} + e_{10t}) + 6 \cdot (e_{12t} + e_{21t}) + 3 \cdot (e_{02t} + e_{20t}) + 10 \cdot (h_{01t} + h_{02t} + h_{10t} + h_{12t} + h_{20t} + h_{21t})\Big]$$

5.3 Constraints

Full planes cannot travel between the same airport when the day changes. Therefore, when $i = j$ then $f_{ijt} = 0$

$$f_{ijt} = 0$$

$$\forall i, j, t \quad i = 0, 1, 2 \quad j = 0, 1, 2 \quad t = 0, 1, 2, 3, 4$$

Similarly, there cannot be cargo hold at an airport i, when that cargo is supposed to be delivered to that same airport i. Hence, when $i = j$, then

$$h_{ijt} = 0$$

$$\forall i, j, t \quad i = 0, 1, 2 \quad j = 0, 1, 2 \quad t = 0, 1, 2, 3, 4$$

Next, at any given time of the cycle there need to be 1200 planes in the system travelling between each day. This constraint is captured by

$$\sum_{i=0}^{2} \sum_{j=0}^{2} f_{ijt} + e_{ijt} = 1200$$

$$\forall t = 0, 1, 2, 3, 4$$

Taking a closer look at the full plane movements, the amount of planes going from airport i with destination j at time t must be equal to the amount of the cargo stored from the day before with destination j added to the amount of cargo that comes in at airport i on that day and has to be delivered to j minus the amount of cargo that the model decides to hold for a delivery on a later time.

$$f_{ijt} = h_{ijt-1} + cargo_{ijt} - h_{ijt}$$

$$\forall i,j,t \quad i=0,1,2 \quad j=0,1,2 \quad t=0,1,2,3,4$$

Additionally, the full planes leaving airport i flying to j at time t cannot exceed the cargo that is actually available at that time t at that airport i.

$$f_{ijt} \leq cargo_{ijt} + h_{ijt-1}$$

$$\forall i,j,t \quad i=0,1,2 \quad j=0,1,2 \quad t=0,1,2,3,4$$

It needs to be ensured that all cargo that is desired to be delivered in a weekly cycle is brought to destination. This constraint makes sure that summing all deliveries from an airport i to j the desired amount is delivered. The desired amount is the sum of all cargo coming into the system at airport i on each day with destination j. Mathematically,

$$\sum_{t=0}^{4} f_{ijt} = \sum_{t=0}^{4} cargo_{ijt}$$

$$\forall i,j \quad i=0,1,2 \quad j=0,1,2$$

The last constraint takes care of tracking the amount of available planes at each time at every airport. The amount of planes flying into a node need to be equal to the amount of planes leaving that node.

$$\sum_{j=0}^{2} f_{ijt} + e_{ijt} = \sum_{j=0}^{2} f_{jit-1} + e_{jit-1}$$

$$\forall i,t \quad i=0,1,2 \quad t=0,1,2,3,4$$

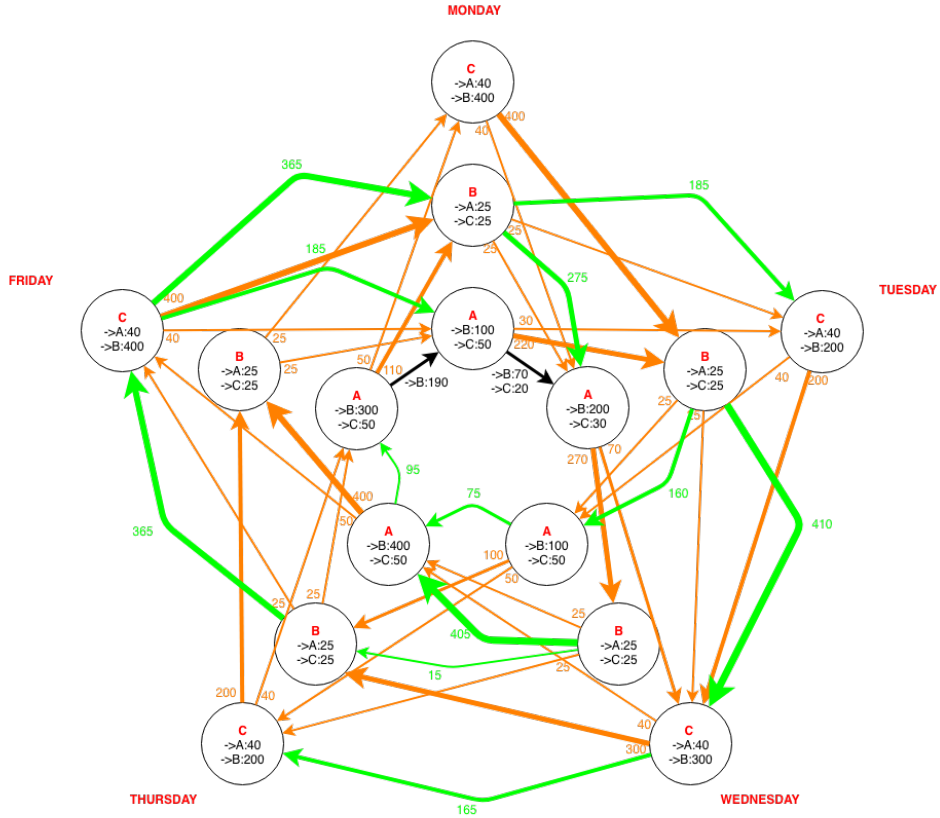
6 Results

Running this optimization model an objective vale of 17,925 is obtained. Figure 1 and Table 4 show the optimal solution found by the optimization model formulated above.

Routes	Monday	Tuesday	Wednesday	Thursday	Friday
A → B	220 0	270 0	100 0	400 0	110 0
A → C	30 0	70 0	50 0	50 0	50 0
B → A	25 275	25 160	25 405	25 0	25 185
B → C	25 185	25 410	25 0	25 365	25 365
C → A	40 0	40 0	40 0	40 0	40 0
C → B	400 0	200 0	300 0	200 0	400 0
A → A	0	0	75	95	0
B → B	0	0	15	0	0
C → C	0	0	165	0	0

Table 4: Copy of Table 1: Optimal assignment of aircrafts for each day on each leg

Figure 1: Cargo and Plane Flow with the width of the arrows representing the amount of flow



7 Analysis

The model found an optimal solution with an objective value of 17,925. The most expansive action is to store cargo in a particular airport on a given day. Looking at Figure 1 it can be seen that 190 loads of cargo are kept at airport A from Friday through Monday. This cargo is supposed to go to airport B. The holding cost of this action alone is 1,900, accounting for more than 10% of the total cost of operating this model. From Monday to Tuesday cargo is hold as well. 70 units are stored on their way to B. 20 are kept with destination C. In total more than 15% of the cost is caused my storing cargo. Empty planes' re-positioning cause the remaining 85% of the cost. Observing Figure 1 it can be seen how there is a lot of demand for deliveries to B but very little demand to ship out of B. This causes the system to have a lot of empty planes flying out of B to ensure plane supply to carry out all needed deliveries on the busy legs from A and C into B.

8 Findings

As it can be seen in Figure 1, a lot of empty planes are flown out of airport B at each day. To reduce the cost of the system significantly Express Air could try to increase the amount of cargo shipped out of B each day. By increasing the amount of cargo that has to be delivered out of airport B from 25 on each day to 100 for each airport, the objective value decreases from 17,925 to 13,050. Increasing that demand to 200 on each leg instead pushes the objective value down to 6,550. On the other hand increasing the demand from each airport to each particular destination by the amount of empty planes exactly, eliminating empty plane transfers, the objective value decreases to only 2300. Without changing the demands, to avoid storing cargo in airports the size of the fleet can be increased. increasing the fleet up to 1390 will decrease the objective value. Anything above that won't affect the performance. The minimum amount of planes for a feasible model is 1130. The best objective value increasing the fleet to 1390 is 15,125. Going back to the change of cargo supply and adding planes to that model, an even lower objective value can be achieved. In fact, having an ideal fleet size of 1390 and having the ideal cargo supply brings the objective value to 0. Another way to decrease cost is to lower the expanse associated with moving empty planes and storing cargo. The route on which the most empty planes are moved is B-¿C. Reducing the cost of that transfer by 1 unit decreases the total cost to 16,600. Similarly, decreasing the holding cost by 1 unit, only decreases the objective value by 280 units. That is much less than decreasing the transportation cost of empty planes.

9 Appendix A

```
\ Model Express Air
\ LP format - for model browsing. Use MPS format to capture full model detail.
Minimize
  7 empty_planes_1_0_0 + 3 empty_planes_2_0_0 + 7 empty_planes_0_1_0
   + 5 empty_planes_2_1_0 + 3 empty_planes_0_2_0 + 5 empty_planes_1_2_0
   + 7 empty_planes_1_0_1 + 3 empty_planes_2_0_1 + 7 empty_planes_0_1_1
   + 5 empty_planes_2_1_1 + 3 empty_planes_0_2_1 + 5 empty_planes_1_2_1
...
   + 10 amount_staying_in_1_going_to_0_on_period_0
   + 10 amount_staying_in_2_going_to_0_on_period_0
   + 10 amount_staying_in_0_going_to_1_on_period_0
   + 10 amount_staying_in_2_going_to_1_on_period_0
...
Subject To
 no_f_0_0_0: full_planes_0_0_0 = 0
 no_f_1_1_0: full_planes_1_1_0 = 0
...
 no_h_0_0_0: amount_staying_in_0_going_to_0_on_period_0 = 0
 no_h_1_1_0: amount_staying_in_1_going_to_1_on_period_0 = 0
...
 1200_planes_at_0: full_planes_0_0_0 + full_planes_1_0_0
   + full_planes_2_0_0 + full_planes_0_1_0 + full_planes_1_1_0
   + full_planes_2_1_0 + full_planes_0_2_0 + full_planes_1_2_0
   + full_planes_2_2_0 + empty_planes_0_0_0 + empty_planes_1_0_0
   + empty_planes_2_0_0 + empty_planes_0_1_0 + empty_planes_1_1_0
   + empty_planes_2_1_0 + empty_planes_0_2_0 + empty_planes_1_2_0
   + empty_planes_2_2_0 = 1200
...
 available_cargo: full_planes_0_1_0
   + amount_staying_in_0_going_to_1_on_period_0
   - amount_staying_in_0_going_to_1_on_period_4 = 100
 ...
 max_amount_of_f_0_1_0: full_planes_0_1_0
   - amount_staying_in_0_going_to_1_on_period_4 <= 100
 ...
 total_cargo_from_0_to_1: full_planes_0_1_0 + full_planes_0_1_1
   + full_planes_0_1_2 + full_planes_0_1_3 + full_planes_0_1_4 = 1100
 ...
 available_planes_at_0: full_planes_0_0_0 + full_planes_0_1_0
   + full_planes_0_2_0 - full_planes_0_0_4 - full_planes_1_0_4
   - full_planes_2_0_4 + empty_planes_0_0_0 + empty_planes_0_1_0
   + empty_planes_0_2_0 - empty_planes_0_0_4 - empty_planes_1_0_4
   - empty_planes_2_0_4 = 0
 ...
Bounds
Generals
 full_planes_0_0_0 full_planes_1_0_0 full_planes_2_0_0 full_planes_0_1_0
 full_planes_1_1_0 full_planes_2_1_0 full_planes_0_2_0 full_planes_1_2_0
 full_planes_2_2_0 full_planes_0_0_1 full_planes_1_0_1 full_planes_2_0_1
 full_planes_0_1_1 full_planes_1_1_1 full_planes_2_1_1 full_planes_0_2_1
 full_planes_1_2_1 full_planes_2_2_1 full_planes_0_0_2 full_planes_1_0_2
 full_planes_2_0_2 full_planes_0_1_2 full_planes_1_1_2 full_planes_2_1_2
 full_planes_0_2_2 full_planes_1_2_2 full_planes_2_2_2 full_planes_0_0_3
 full_planes_1_0_3 full_planes_2_0_3 full_planes_0_1_3 full_planes_1_1_3
 full_planes_2_1_3 full_planes_0_2_3 full_planes_1_2_3 full_planes_2_2_3
 full_planes_0_0_4 full_planes_1_0_4 full_planes_2_0_4 full_planes_0_1_4
 full_planes_1_1_4 full_planes_2_1_4 full_planes_0_2_4 full_planes_1_2_4
 full_planes_2_2_4 empty_planes_0_0_0 empty_planes_1_0_0 empty_planes_2_0_0
 empty_planes_0_1_0 empty_planes_1_1_0 empty_planes_2_1_0
 empty_planes_0_2_0 empty_planes_1_2_0 empty_planes_2_2_0
 empty_planes_0_0_1 empty_planes_1_0_1 empty_planes_2_0_1
```

Figure 2: Shortened LP file outlining output to Gurobi

10 Appendix B

CARGO OPERATIONS OF AIR EXPRESS

In [1]:
```
# Import data and packages


from gurobi import *

cargo = [[[0 for t in range(5)]for i in range(3)]for j in range(3)]

cargo[0][1][0]=100
cargo[0][2][0]=50
cargo[1][0][0]=25
cargo[1][2][0]=25
cargo[2][0][0]=40
cargo[2][1][0]=400

cargo[0][1][1]=200
cargo[0][2][1]=50
cargo[1][0][1]=25
cargo[1][2][1]=25
cargo[2][0][1]=40
cargo[2][1][1]=200

cargo[0][1][2]=100
cargo[0][2][2]=50
cargo[1][0][2]=25
cargo[1][2][2]=25
cargo[2][0][2]=40
cargo[2][1][2]=300

cargo[0][1][3]=400
cargo[0][2][3]=50
cargo[1][0][3]=25
cargo[1][2][3]=25
cargo[2][0][3]=40
cargo[2][1][3]=200

cargo[0][1][4]=300
cargo[0][2][4]=50
cargo[1][0][4]=25
cargo[1][2][4]=25
cargo[2][0][4]=40
cargo[2][1][4]=400

# define model

model = Model("Express Air")
```

```
Academic license - for non-commercial use only
```

Parameters

In [2]:
```
COST_AB = 7
COST_BC = 6
COST_AC = 3
COST_HOLD = 10
FLEET = 1200
```

Decision Variables

In [3]:
```
#full plane movements

f = [[[0 for t in range(5)]for j in range(3)]for i in range(3)]

for t in range(5):
    for j in range(3):
        for i in range(3):
            f[i][j][t]=model.addVar(vtype = GRB.INTEGER,
                                    name = "full_planes_"+ str(i)+ "_" + str

model.update()

#empty plane movements

e = [[[0 for t in range(5)]for j in range(3)]for i in range(3)]

for t in range(5):
    for j in range(3):
        for i in range(3):
            e[i][j][t]=model.addVar(vtype = GRB.INTEGER,
                                    name = "empty_planes_" + str(i)+ "_" + s

model.update()

#cargos remaining in airports

h = [[[0 for t in range(5)]for j in range(3)]for i in range(3)]

for t in range(5):
    for j in range(3):
        for i in range(3):
            h[i][j][t]=model.addVar(vtype = GRB.INTEGER,
                                    name = "amount_staying_in_" + str(i)+ "_
            
model.update()
```

Objective Function

In [4]:
```
objective_function = LinExpr()

for t in range(5):
    objective_function+= COST_AB*(e[0][1][t]+e[1][0][t])+COST_BC*(e[1][2][t]

model.setObjective(objective_function,GRB.MINIMIZE)

model.update()
```

Constraints

In [5]:

```python
# no full planes from node to node

for t in range(5):
    for i in range(3):
        for j in range(3):
            if i==j:
                model.addConstr(lhs = f[i][j][t],
                                sense = GRB.EQUAL,
                                rhs = 0,
                                name = "no_f_" + str(i) + "_" + str(j) + "_'

# no holding from node to node

for t in range(5):
    for i in range(3):
        for j in range(3):
            if i==j:
                model.addConstr(lhs = h[i][j][t],
                                sense = GRB.EQUAL,
                                rhs = 0,
                                name = "no_h_" + str(i) + "_" + str(j) + "_'


# 1200 planes constraint

for t in range(5):
    lhs = LinExpr()
    for i in range(3):
        for j in range(3):
            lhs+=f[i][j][t]+e[i][j][t]
    model.addConstr(lhs = lhs,
                    sense = GRB.EQUAL,
                    rhs = FLEET,
                    name = "1200_planes_at_" + str(t))


# cargo entering node = cargo leaving node

for t in range(5):
    for i in range(3):
        for j in range(3):
            lhs = f[i][j][t]
            rhs = h[i][j][t-1] + cargo[i][j][t] - h[i][j][t]
            model.addConstr(lhs = lhs,
                            sense = GRB.EQUAL,
                            rhs = rhs,
                            name = "available_cargo" )

model.update()

# node supply requirements

for t in range(5):
    for i in range(3):
        for j in range(3):
```

```
            model.addConstr(lhs = f[i][j][t],
                            sense = GRB.LESS_EQUAL,
                            rhs = cargo[i][j][t] + h[i][j][t-1],
                            name = "max_amount_of_f_" + str(i) + "_" + str(j


# total cargo from i to j

for i in range(3):
    for j in range(3):
        lhs=LinExpr()
        rhs = LinExpr()
        for t in range(5):
            lhs+=f[i][j][t]
            rhs+=cargo[i][j][t]
        model.addConstr(lhs = lhs,
                        sense = GRB.EQUAL,
                        rhs = rhs,
                        name = "total_cargo_from_" + str(i) + " to " + str(j


# constrain amount of available planes at each airport at each time

for t in range(5):
    for i in range(3):
        lhs=LinExpr()
        rhs=LinExpr()
        for j in range(3):
            lhs+=f[i][j][t]+e[i][j][t]
            rhs+=f[j][i][t-1]+e[j][i][t-1]
        model.addConstr(lhs = lhs,
                        sense = GRB.EQUAL,
                        rhs = rhs,
                        name = "available_planes_at_" + str(i))



model.update()
```

OPTIMIZE

In [6]:
```python
model.write("ExpressAirV4.lp")

model.optimize()
print("\nOptimal Objective: " + str(model.ObjVal))

#extract decision variables
print("\nOptimal Solution:")
allVars = model.getVars()
for var in allVars:
    if var.x != 0:
        print(var.varName + " " + str(var.x))
```

```
Warning: constraint name "total_cargo_from_0 to 0" has a space
Optimize a model with 149 rows, 135 columns and 570 nonzeros
Variable types: 0 continuous, 135 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [3e+00, 1e+01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+01, 2e+03]
Presolve removed 99 rows and 54 columns
Presolve time: 0.00s
Presolved: 50 rows, 81 columns, 315 nonzeros
Variable types: 0 continuous, 81 integer (0 binary)

Root relaxation: objective 1.792500e+04, 43 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node T
ime

*    0     0               0    17925.000000 17925.0000  0.00%     -    0
s

Explored 0 nodes (43 simplex iterations) in 0.02 seconds
Thread count was 8 (of 8 available processors)

Solution count 1: 17925

Optimal solution found (tolerance 1.00e-04)
Best objective 1.792500000000e+04, best bound 1.792500000000e+04, gap 0.0
000%

Optimal Objective: 17925.0

Optimal Solution:
full_planes_1_0_0 25.0
full_planes_2_0_0 40.0
full_planes_0_1_0 220.0
full_planes_2_1_0 400.0
full_planes_0_2_0 30.0
full_planes_1_2_0 25.0
full_planes_1_0_1 25.0
full_planes_2_0_1 40.0
```

```
full_planes_0_1_1 270.0
full_planes_2_1_1 200.0
full_planes_0_2_1 70.0
full_planes_1_2_1 25.0
full_planes_1_0_2 25.0
full_planes_2_0_2 40.0
full_planes_0_1_2 100.0
full_planes_2_1_2 300.0
full_planes_0_2_2 50.0
full_planes_1_2_2 25.0
full_planes_1_0_3 25.0
full_planes_2_0_3 40.0
full_planes_0_1_3 400.0
full_planes_2_1_3 200.0
full_planes_0_2_3 50.0
full_planes_1_2_3 25.0
full_planes_1_0_4 25.0
full_planes_2_0_4 40.0
full_planes_0_1_4 110.0
full_planes_2_1_4 400.0
full_planes_0_2_4 50.0
full_planes_1_2_4 25.0
empty_planes_1_0_0 275.0
empty_planes_1_2_0 185.0
empty_planes_1_0_1 160.0
empty_planes_1_2_1 410.0
empty_planes_0_0_2 75.0
empty_planes_1_0_2 405.0
empty_planes_1_1_2 15.0
empty_planes_2_2_2 165.0
empty_planes_0_0_3 95.0
empty_planes_1_2_3 365.0
empty_planes_1_0_4 185.0
empty_planes_1_2_4 365.0
amount_staying_in_0_going_to_1_on_period_0 70.0
amount_staying_in_0_going_to_2_on_period_0 20.0
amount_staying_in_0_going_to_1_on_period_4 190.0
```

In [ ]: