Optimization Methods Project
Cargo Operations of Express Air

Aya Chehabi (ac2532@cornell.edu), Naman Makkar (nbm49@cornell.edu)

December 10, 2022

Contents

List of Figures

List of Tables

1 Background

In this section, we will outline Express Air's activities, the main issue the business is encountering, the data they have provided us with and the way we plan to solve the problem.

1.1 Business description

Express Air is a cargo delivery business that operates between three airports, Airport A, Airport B, and Airport C. They specialize in transporting cargo efficiently and reliably between these airports. They have a fleet of 1200 aircraft, which is equipped to handle a wide variety of cargo and allows them to handle a large volume of cargo. Their system receives new cargoes to be delivered between these airports every weekday. With Express Air Cargo, the clients can trust that the cargo will be in good hands from start to finish.

Express Air is committed to providing reliable, efficient cargo delivery services. The experienced team, state-of-the-art aircraft, and commitment to excellence make them the trusted choice for all of the cargo transportation needs.

1.2 Problem overview

Express Air's cargo operator carefully assesses the cargo that arrives in the system each weekday and the cargoes that were not delivered on the previous days to decide which cargo will be delivered on that day. The cargo operator also determines the aircraft movement: how many aircraft will be used for a particular origin-destination shipment, how many aircraft will be repositioned from airport i to airport j, and how many aircraft will not be used and will stay on the ground on any given weekday.

We will have to determine the optimal plan for delivering the cargoes to their respective destinations. This is a cyclic scheduling problem. We will take into account the amount of cargo that was not delivered the previous day, the new cargoes entering the system, as well as the number of aircraft available at each airport on that day. In addition to these requirements, we also have to take into account aircraft repositioning to help balance the flow of inbound and outbound cargo at each airport. This will allow Express Air to maximize the efficiency of their operations. Our objective is to minimize the overall costs of the empty aircraft repositioning between airports, and most importantly the costs of holding cargoes in the warehouse instead of shipping them.

1.3 Data description

Express Air has provided us with the following data.

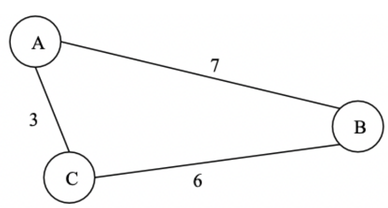


Figure 1: Empty repositioning costs among different airports

Figure 1 shows that aircraft reposition from airport A to B or from airport B to A will both cost 7 unit, repositioning an aircraft from airport A to C or from airport C to A will cost 3 units, and finally repositioning an aircraft from airport B to C or from airport C to B will cost 6 units. These values will be useful in the objective function of our integer linear program. Table 1 (found below) shows the number of new cargoes arriving into the system for each day of the week, respective of their origin-destination airport shipment itinerary. For instance , each week day, 50 cargoes arrive at airport B, half of them has to be shipped to airport A while the other half has to be shipped to airport C. Another example is on Thursday 200 cargoes arrive at airport C and have to be delivered to airport B.

Origin-destination \ Day	Monday	Tuesday	Wednesday	Thursday	Friday
A-B	100	200	100	400	300
A-C	50	50	50	50	50
B-A	25	25	25	25	25
B-C	25	25	25	25	25
C-A	40	40	40	40	40
C-B	400	200	300	200	400

Table 1: Amounts of cargo (in aircraft loads) arriving into the system on each day that need to be carried between each origin-destination airport

1.4 Our goal

The objective of this paper is to find an optimized plan for the operation of the fleet of aircraft and cargoes received over a week for Express Air. The optimized aircraft schedule will include details such as the routes that the aircraft and cargoes will fly, the airports at which it will arrive and depart, and the day at which these events will take place. The aircraft schedule is an important tool for coordinating the operation of Express air and ensuring that cargoes are transported efficiently and reliably. The development of the schedule involves a number of factors, including the availability of aircraft, the demand for shipments on particular routes, and the need to maintain an efficient weekly cycle. To do so, an integer linear program will be developed, and the software Gurobi will compute the optimized schedule.

This schedule will be a critical component of Express Air's operations as it is important for the airline to maintain a flexible and adaptable schedule to meet the changing needs of all its cargo customers.

2 Optimization model overview

A model using integer linear programming has been developed to optimize the schedule for the aircraft and cargo fleet. The goal of the model is to satisfy all delivery demands while considering the problem's objective. Thus, in this section, we will introduce the integer linear program, in mathematical and coded form, that solves the issue stated in the previous section, then we will present the solution provided by Gurobi.

2.1 Mathematical details

2.1.1 Assumptions

In order to formulate and develop the model, some assumptions were made. These assumptions provide a basis for the model and help to ensure that it accurately reflects the situation being modeled.

- Assumption 1: The cost per day per cargo held on the ground is 10, therefore we can assume that the cost for cargo held on the ground on Friday is 30 since the cargo cannot be moved until Monday and has to be held for 3 days.

- Assumption 2: The cargoes are delivered directly from their origin location to destination location. For instance, a cargo needed to be delivered from airport A to B **cannot** go through airport C to be delivered to airport B.

- Assumption 3: There are no induced cost if an aircraft is not used and stays on the ground on any given day in any airport.

- Assumption 4: The cargo arriving on Monday to any node includes the cargoes held on the ground on Friday in addition to the amount of cargoes coming into the system on Monday given in Table 1. In order to account for this we have to write wraparound constraints for Monday and Friday.

- Assumption 5: Aircraft Flow Balance - We make the assumption that on any given day t the sum of the number of cargoes transferred from node i to all other nodes + the sum of the number of empty repositionings from node i to all other nodes + the number of idle aircrafts at node i should be exactly equal to the sum of the number of cargoes transferred from all other nodes to node i on day t-1 + the sum of all empty repositionings from all other nodes to node i + the number of idle aircrafts at node i on day t-1.
- Assumption 6: Cargo Flow Balance - We make the assumption that the amount of cargo available for transporting from node i to node j on day t is exactly equal to the amount of cargo available for transporting from node i to node j on day t-1 - the amount of cargo that was transported from node i to node j on day t-1 + the amount of cargo moving in to the system for being transported from node i to node j on day t as given in the data in Table 1.
- Assumption 7: The unit of the costs is 10,000.00 United States Dollars.
- Assumption 8: The schedule we will suggest will not have a 'start' day, it will be part of a cyclic schedule.
- Assumption 9: The travel time between each airport i to airport j for any aircraft is a full day: if an aircraft is used to ship certain amount of cargo from airport i to j during day t, then the aircraft becomes available at airport j at the beginning of day t+1.

2.1.2 Notations

The airports $\{A, B, C\}$ have been denoted by the nodes $\{1, 2, 3\}$ respectively, and each day of the week has been denoted by $\{1, 2, 3, 4, 5\}$ respective to Monday, Tuesday, Wednesday, Thursday, and Friday.

2.2 Integer Linear Program

2.2.1 Data

The data Express Air has provided us with will be used as the following:

- c_{ij} - The cost of empty repositioning from node i to node j. $(i \neq j)$
- D_{ijt} - The amount of cargo coming into node i to be transferred to node j on day t according to the data provided. $(i \neq j)$

2.2.2 Decision variables

We made use of the following decision variables -

- x_{ijt} - The number of planes transporting cargo from node i to node j on day t. We use variable 'shipment_on_dayt_fromi_toj' in the code to denote this decision variable. Where $i \neq j$
- y_{ijt} - The number of planes that need to be repositioned from node i to node j on day t without any cargo. We use the variable 'repositioning_dayt_fromi_toj' in the code to denote this decision variable. Where $i \neq j$
- u_{ijt} - The amount of cargo that needs to be transferred from node i to node j on day t. We use the variable 'cargos_needed_to_be_shipped_dayt_fromi_toj' in the code to denote this decision variable. Where $i \neq j$
- z_{it} - The number of planes idle at node i on day t. This decision variable is denoted by 'repositioning_dayt_fromi_toj' in the code. Where $i = j$ and we get the number of aircraft idle at node i on day t.

2.2.3 Objective function

The objective function has been written to minimize the total cost of empty repositioning and cargo not transported. Since the cost of holding one full aircraft load of cargo on the ground is 10 per day, we account the cost for holding a full load of cargo on the ground on Friday as 30.

$$\min \sum_{t=1}^{4} \sum_{i=1}^{3} \sum_{j=1}^{3} (c_{ij} y_{ijt} + 10(u_{ijt} - xijt)) + \sum_{i=1}^{3} \sum_{j=1}^{3} (c_{ij} y_{ij5} + 30(u_{ij5} - x_{ij5}))$$

2.2.4 Constraints

We make use of the following constraints -

- Total Aircrafts Constraint - $\sum_{i=1}^{3}\sum_{j=1}^{3} x_{ijt} + y_{ijt} + z_{it} = 1200 \ \forall t \in \{1,2,3,4,5\}$
- Aircraft Flow Balance Constraints -
  - $\sum_{j=1}^{3}(x_{ijt} + y_{ijt}) + z_{it} = \sum_{j=1}^{3}(x_{ji(t-1)} + y_{ji(t-1)}) + z_{i(t-1)} \ \forall i \in \{1,2,3\}, t \in \{2,3,4,5\}$
  - $\sum_{j=1}^{3}(x_{ij1} + y_{ij1}) + z_{i1} = \sum_{j=1}^{3}(x_{ji5} + y_{ji5}) + z_{i5} \ \forall i \in \{1,2,3\}$
- Cargo Flow Balance Constraints -
  - $u_{ijt} = u_{ij(t-1)} - x_{ij(t-1)+D_{ijt}} \ \forall i,j \in \{1,2,3\}, t \in \{2,3,4,5\}$ where $i \neq j$
  - $u_{ij1} = u_{ij5} - x_{ij5+D_{ij1}} \ \forall i,j \in \{1,2,3\}$
  - $x_{ijt} \leq u_{ijt} \ \forall i,j,t$

$x_{ijt}, y_{ijt}, u_{ijt}, z_{it} \geq 0 \ x_{ijt}, y_{ijt}, u_{ijt}, z_{it} \in \mathbf{N} \ \forall i \in \{1,2,3\}, \forall j \in \{1,2,3\} \ (i \neq j), \forall t \in \{1,2,3,4,5\}$

2.2.5 Python Code

Ultimately, we translated the above linear program into the code found in Appendix A.

2.3 Gurobi Solution

The solution given by Gurobi optimizer (found in Appendix B) has been translated into the tables below to simplify the understanding of the solution. This solution gives an objective value of 21,725.00, i.e. \$217,250,000.00 .

Origin-Destination of the aircraft reposition	Monday	Tuesday	Wednesday	Thursday	Friday
A – B	0	0	0	0	0
A – C	0	0	0	0	0
B – A	275	85	495	170	0
B – C	165	265	165	180	550
C – A	0	0	0	0	0
C – B	0	0	0	0	0

Table 2: Number of aircraft repositioned from origin airport to destination airport on any given weekday

	Monday	Tuesday	Wednesday	Thursday	Friday
A	0	0	0	110	0
B	20	220	0	0	0
C	0	0	0	0	0

Table 3: Number of aircraft staying on the ground in the airport on any given weekday

Origin-Destination of the shipment	Monday	Tuesday	Wednesday	Thursday	Friday
A – B	15	290	100	400	295
A – C	50	50	50	50	50
B – A	25	25	25	25	25
B – C	25	25	25	25	25
C – A	40	40	40	40	40
C – B	585	200	300	200	215

Table 4: Number of aircraft shipping cargo from origin-airport to destination-airport on any given weekday

Origin-Destination of the total cargo needed to be shipped	Monday	Tuesday	Wednesday	Thursday	Friday
A – B	105	290	100	400	300
A – C	50	50	50	50	50
B – A	25	25	25	25	25
B – C	25	25	25	25	25
C – A	40	40	40	40	40
C – B	585	200	300	200	400

Table 5: Total number of cargo that have to be delivered at the beginning of each day in each airport

3 Analysis of the results and findings

In this section, we will prove that our proposed fleet schedule satisfies all the requirements imposed by Express Air, and analyze the results.

3.1 Graph Visualization of the solution

Please refer to Appendix C to visualize the flow of the empty aircraft between airports. On Friday 550 airplanes are repositioned from airport B to C which means that these airplanes are found available on Monday in airport C. The flow repeats itself each week, thus, the graph represents a cycle from one week to another in our schedule.

3.2 Arguments proving that the proposed schedule is sensible

Let us ensure that the results are sensible and practicable such that Express Air can operate their aircraft and cargo fleet using Gurobi optimizer's solution. To do so, we will present the exact fleet schedule and make sure that the results are sensible.

3.2.1 Total aircraft constraint

We have summed the columns of the three tables above. That is adding the number of airplanes leaving airports A, B and C (that is airplanes being repositioned and airplanes used to ship cargo), to the number airplanes staying on the ground in these airports for each day of the week (Monday, Tuesday, Wednesday, Thursday and Friday), and the results are 1200 for all summations, which validates our first constraint. Taking as an example Tuesday, the list of airplanes on that day is the following:

- In Table 2: 85 + 265 planes leaving airport B to be repositioned to airplane A and C respectively on Tuesday.

- In Table 3: 220 airplanes staying idle at airport B on Tuesday.
- In Table 4: 290 airplanes are shipping cargo from airport A to B on Tuesday, 50 from A to C, 25 from B to A, 25 from B to C, 40 from C to A and finally, 200 from C to B. This holds a total of 1200 airplanes (either shipping cargo, being repositioned or staying idle).

Below, is the excel table calculating these summations for each day of the week:

	Monday	Tuesday	Wednesday	Thursday	Friday
Total number of aircraft	1200	1200	1200	1200	1200

Table 6: Total number of aircraft operating on any given day

3.2.2 Aircraft flow balance constraint

Now, let us prove that the proposed schedule satisfy the aircraft flow balance, that is if an aircraft moves from airport i to j on day t for a shipment or a reposition, then the aircraft has to be available at airport j at the beginning of day t+1.

Since Express Air needs a weekly cycle to manage its fleet, the total number of aircraft that flies into a particular airport or stays at that airport at the end of a week should be equal to the total number of aircraft that leaves this airport or stays at the airport at the beginning of the next week. Thus we have to make sure that the number of airplanes at the end of Friday (after all airplanes have finished their shipment and reposition) needs to be equal to the number of airplanes available on Monday morning (before the beginning of any movement). To do so, we will verify for each airport that the airplanes' movement at the end of Friday matches the way airplanes were used on Monday.

Using Table 2, 3 and 4 we can analyse the aircraft flow constraints. Looking at the aircraft flow on Monday (i.e the total aircraft transporting cargo out of node i, the total empty repositionings out of node i and the number of idle aircraft at node i) we can observe that $\sum_{j=1}^{3}(x_{Aj1} + y_{Aj1}) + z_{A1} = \sum_{j=1}^{3}(x_{jA5} + y_{jA5}) + z_{A5} = 65$ therefore there exists a flow balance at airport A between Friday and Monday. Similarly, $\sum_{j=1}^{3}(x_{Bj1} + y_{Bj1}) + z_{B1} = \sum_{j=1}^{3}(x_{jB5} + y_{jB5}) + z_{B5} = 510$ and $\sum_{j=1}^{3}(x_{Cj1} + y_{Cj1}) + z_{C1} = \sum_{j=1}^{3}(x_{jC5} + y_{jC5}) + z_{C5} = 625$.
Let us carefully examine these calculations by computing the number of airplanes available at the beginning of Monday at each airport:

- For airport A: shipment B - A uses 25 airplanes and shipment C - A uses 40 airplanes, this leads to a total of 65 airplanes available at the beginning of Monday
- For airport B: shipment A - B uses 295 airplanes and shipment C - B uses 215 airplanes, this leads to a total of 510 airplanes available at the beginning of Monday
- For airport C: shipment A - C uses 50 airplanes and shipment B - C uses 25 airplanes, 550 airplanes are repositioned from airport B to C, this leads to a total of 615 airplanes available at the beginning of Monday

Now, let us ensure that these numbers match the number of airplanes used on Monday at each airport:

- For airport A: shipment A - B uses 15 airplanes and shipment A - C uses 50 airplanes, this leads to a total of 65 airplanes used on Monday
- For airport B: shipment B - A uses 25 airplanes, shipment B - C uses 25 airplanes, 275 airplanes are repositioned from B to A and 165 from B to C, we also have to take into account the 20 airplanes staying idle. This leads to a total of 510 airplanes used on Monday.
- For airport C: shipment C - A uses 40 airplanes and shipment C - A uses 585 airplanes, this leads to a total of 625 airplanes available at the beginning of Monday

As we can see, the numbers match exactly. We can confirm that the same schedule can be used weekly indefinitely as long as the requirements do not change.

Similarly, we calculate the flow balance for the remaining days using an Excel sheet. The results are presented in table 7 and 8 below. We proceed to a pairwise comparison between the two tables below to make sure that the values match. Excel's conditional formatting shows that all the values match exactly as they appear with a green background. **Therefore the solution satisfies all of the aircraft flow balance constraints.**

Number of aircraft available at the beginning of any day					
	Monday	Tuesday	Wednesday	Thursday	Friday
A	65	340	150	560	345
B	510	620	710	400	600
C	625	240	340	240	255
Total	1200	1200	1200	1200	1200

Table 7: Total number of aircraft available at the beginning of the day in each airport

Number of aircraft leaving the airport at the end of the day					
	Monday	Tuesday	Wednesday	Thursday	Friday
A	65	340	150	560	345
B	510	620	710	400	600
C	625	240	340	240	255
Total	1200	1200	1200	1200	1200

Table 8: Total number of aircraft used in each airport

3.2.3 Cargo flow balance constraint

In Table 5, we list the total amount of cargo that needs to be delivered from one airport to another. If we have a look at the total amount of cargo on Monday, we can observe that $u_{AB1} = 105, u_{AC1} = 50, u_{BA1} = 25, u_{BC1} = 25, u_{CA1} = 40, u_{CB1} = 585$ Having a look at Table 1 and Table 4 simultaneously, we can verify that the cargo flow balance constraint has been followed. Since $u_{AB1} = u_{AB5} - x_{AB5} + D_{AB1} = 300 - 295 + 100 = 105, u_{AC1} = u_{AC5} - x_{AC5} + D_{AC1} = 50 - 50 + 50 = 50, u_{BA1} = u_{BA5} - x_{BA5} + D_{BA1} = 25 - 25 + 25 = 25, u_{BC1} = u_{BC5} - x_{BC5} + D_{BC1} = 25 - 25 + 25 = 25, u_{CA1} = u_{CA5} - x_{CA5} + D_{CA1} = 40 - 40 + 40 = 40, u_{CB1} = u_{CB5} - x_{CB5} + D_{CB1} = 400 - 215 + 400 = 585$

Similarly, we can analyse the cargo flow balance for Tuesday - $u_{AB2} = u_{AB1} - x_{AB1} + D_{AB2} = 105 - 15 + 200 = 290, u_{AC2} = u_{AC1} - x_{AC1} + D_{AC2} = 50 - 50 + 50 = 50, u_{BA2} = u_{BA1} - x_{BA1} + D_{BA2} = 25 - 25 + 25 = 25, u_{BC2} = u_{BC1} - x_{BC1} + D_{BC2} = 25 - 25 + 25 = 25, u_{CA2} = u_{CA1} - x_{CA1} + D_{CA2} = 40 - 40 + 40 = 40, u_{CB2} = u_{CB1} - x_{CB1} + D_{CB1} = 585 - 585 + 200 = 200$ and a similar analysis can be carried out for Wednesday, Thursday and Friday.

Looking at Table 4 and Table 5, we can make an interesting observation for Tuesday, Wednesday and Thursday, which is the number of aircraft transporting cargo between two airports is exactly equal to the amount of cargo needed to be delivered on that day at those airports. That is to say: $x_{ijt} = u_{ijt}$ $\forall i, j \in \{1, 2, 3\}, t \in \{2, 3, 4\}$

Similarly, looking at the values in Table 4 and Table 5 for Monday we can observe that the total amount of cargo shipped from A-B is less than the total amount of cargo ready to be shipped from A-B. Similarly, on Friday, the total amount of cargo transported from A-B is less than the total cargo required to be shipped and the total cargo transported from C-B is less than the cargo required to be transported. We have observed that: $x_{ijt} \leq u_{ijt} \forall i, j \in \{1, 2, 3\}, t \in \{1, 2, 3, 5\}$

Therefore, we can conclude that the solution satisfies all the cargo flow balance constraints.

3.3 Results Summary

3.3.1 The Proposed Fleet Schedule

For ease of reference, we have put together the following fleet schedule in the form of tables that Express Air can conveniently and handily consult. This schedule can be found in Appendix D.

3.3.2 Interesting findings and interpretation

Looking at Table 2 and Appendix C an interesting observation can be made that all of the empty repositioning occurs between B-A and B-C and majority of all the cargo transportation is carried out between the pairs A-B and C-B. This is an additional piece of circumstantial evidence that the solution follows aircraft flow constraints since the empty repositioning costs between airport A and C are the lowest but there is no empty repositioning between these two airports.
Moreover, we can notice that the total number of cargo left in the warehouse and not transported on the day they arrived is the highest on Friday specifically for A-B and C-B. This is particularly interesting because it costs three times more to keep a cargo from Friday to Monday than any other pair of days (t and t+1) as keeping a cargo in the warehouse for one day cost $100,000.00 (10*$10,000), but since we have to account for the weekend as well keeping a cargo from Friday to Monday will cost $300,000.00. This follows the intuition that repositioning multiple aircraft to complete a delivery for these cargoes would cost more than keeping the cargoes over the weekend.

3.4 Further actions

As previously proved, our model outputs a sensible solution to the problem statement. Our team has suggestions to Express Air in order to improve their fleet operations. First of all, Express Air may be interested in making the delivery times shorter. This process will induce higher costs but it can also attract new clients which may be more profitable for Express Air. The second option to make faster deliveries more profitable is to increase the cargo delivery prices to clients. Moreover, it would also be interesting to consider investing in new aircraft. We will have to analyze the difference in profit in the long term between our current schedule and the one with the extra aircraft minus the costs induced by buying the new aircraft. If this difference is negative, then it would be more profitable to buy new airplanes in the long term.

4 Conclusion

Overall, the problem has a feasible solution that has been presented in this paper. All of the requirements imposed by Express Air have been included in the development of the linear program and satisfied by the proposed schedule. The same fleet schedule can be used weekly indefinitely as long as the requirements and data do not change. However, the schedule may be adjusted periodically in the future to reflect changes in delivery demand (i.e. number of cargoes entering the system), number of aircraft available, changing costs or other factors.

A Python Code

In [1]:
```
from gurobipy import *
```

In [2]:
```
# initialize the data for the problem

no_days = 5
no_airports = 3

data = [ [100,200,100,400,300],
        [50,50,50,50,50],
        [25,25,25,25,25],
        [25,25,25,25,25],
        [40,40,40,40,40],
        [400,200,300,200,400]
    ]

reposition_costs = [[[0,7,3],
                    [7,0,6],
                    [3,6,0]]]*5
print(reposition_costs)
warehouse_costs = 10


new_cargoes = [ [[ 0 for j in range( no_airports ) ] for i in range( no_airports ) ] for k in range( no_days )


new_cargoes = [
    [ [0, 100, 50], [25, 0, 25], [40, 400, 0] ], #Monday - Day 0
    [ [0, 200, 50], [25, 0 ,25], [40, 200, 0] ], #Tuesday - Day 1
    [ [0, 100, 50], [25, 0 ,25], [40, 300, 0] ], #Wednesday - Day 2
    [ [0, 400, 50], [25, 0 ,25], [40, 200, 0] ],  #Thursday - Day 3
    [ [0, 300, 50], [25, 0 ,25], [40, 400, 0] ],  #Friday - Day 4
]
```

```
[[[0, 7, 3], [7, 0, 6], [3, 6, 0]], [[0, 7, 3], [7, 0, 6], [3, 6, 0]], [[0, 7, 3], [7, 0, 6], [3, 6, 0]], [[0,
7, 3], [7, 0, 6], [3, 6, 0]], [[0, 7, 3], [7, 0, 6], [3, 6, 0]]]
```

Airport 1 is A, Airport 2 is B, Airport 3 is C

Monday is day 1, Tuesday is day 2, Wednesday is day 3, Thursday is day 4, Friday is day 5

In [3]:
```
# create an empty model
myModel = Model( "project" )

# create decision variables

# number of aircrafts repositioned from airport i to airport j on day t
reposition = [ [[ 0 for j in range( no_airports ) ] for i in range( no_airports ) ] for k in range( no_days ) ]
for k in range( no_days ):
    for i in range( no_airports ):
        for j in range ( no_airports ):
            curVar = myModel.addVar( vtype = GRB.INTEGER , name = "repositioning_day" + str(k+1) + "_from" + st
            reposition[ k ][ i ][ j ] = curVar

# number of cargoes shipped from airport i to j on day t
ship = [ [[ 0 for j in range( no_airports ) ] for i in range( no_airports ) ] for k in range( no_days ) ]
for k in range( no_days ):
    for i in range( no_airports ):
        for j in range ( no_airports ):
            curVar = myModel.addVar( vtype = GRB.INTEGER , name = "shipment_on_day" + str(k+1) + "_from" + str(
            ship[ k ][ i ][ j ] = curVar


# number of cargoes that have to be shipped from airport i to airport j on day t
x = [ [[ 0 for j in range( no_airports ) ] for i in range( no_airports ) ] for k in range( no_days ) ]
for k in range( no_days ):
    for i in range( no_airports ):
        for j in range ( no_airports ):
            curVar = myModel.addVar( vtype = GRB.INTEGER , name = "cargoes_needed_to_be_shipped_day" + str(k+1)
            x[ k ][ i ][ j ] = curVar

# number of aircrafts available at airport i on day t
# availability = [ [ 0 for i in range( no_airports ) ] for k in range( no_days ) ]
# for k in range( no_days ):
#     for i in range( no_airports ):
#         curVar = myModel.addVar( vtype = GRB.INTEGER , name = "available_day" + str(k+1) + "_airport" + s
#         availability[ k ][ i ] = curVar

# register all decision variables into model
myModel.update()
```

```
Set parameter Username
Academic license - for non-commercial use only - expires 2023-10-08
```

In [4]:
```
# create objective function
objExpr = LinExpr()
```

```
for k in range( no_days ):
    for i in range( no_airports ):
        for j in range( no_airports ):
            if i != j:
                if k == 4:
                    objExpr += reposition[ k ][ i ][ j ] * reposition_costs[ k ][ i ][ j ] + (x[ k ][ i ][ j ]
                else:
                    objExpr += reposition[ k ][ i ][ j ] * reposition_costs[ k ][ i ][ j ] + (x[ k ][ i ][ j ]

myModel.setObjective( objExpr , GRB.MINIMIZE )
```

In [5]:
```
# create constraints that ensures there are 1200 aircrafts running at any given day
for k in range(no_days):
    constExpr = LinExpr()
    for i in range(no_airports):
        for j in range(no_airports):
            if i != j :
                constExpr += ship[ k ][ i ][ j ] + reposition[ k ][ i ][ j ]
            else:
                constExpr += reposition[ k ][ i ][ j ]
    myModel.addConstr( lhs = constExpr , sense = GRB.EQUAL , rhs = 1200 , name = "aircraftno_day" + str(k+1))

# constraint that ensures the number of aircrafts leaving an airport or being held on the ground
# on any given day is the same as the number of aircrafts coming into the airport or being held
# on the ground the previous day
for k in range(no_days):
    for i in range(no_airports):
        constExpr = LinExpr()
        for j in range(no_airports):
            if i != j :
                if k == 0 :
                    constExpr += ship[ k ][ i ][ j ] - ship[ 4 ][ j ][ i ]
                    constExpr+= reposition[ k ][ i ][ j ] - reposition[ 4 ][ j ][ i ]
                else:
                    constExpr += ship[ k ][ i ][ j ] - ship[ k-1 ][ j ][ i ]
                    constExpr+= reposition[ k ][ i ][ j ] - reposition[ k-1 ][ j ][ i ]
            else:
                if k == 0 :
                    constExpr+= reposition[ k ][ i ][ j ] - reposition[ 4 ][ j ][ i ]
                else:
                    constExpr+= reposition[ k ][ i ][ j ] - reposition[ k-1 ][ j ][ i ]
        myModel.addConstr( lhs = constExpr , sense = GRB.EQUAL , rhs = 0 , name = "aircraftflow_airport_" + str


# create constraint that ensure the cargoes that need to be shipped from airport i to j on day t
# is the same as the cargoes that were available the previous day, minus the cargoes that were shipped
# and put in the warehouse the previous days, plus the new cargoes
for k in range(no_days):
    for i in range(no_airports):
        for j in range(no_airports):
            constExpr = LinExpr()
            if i != j :
                if k == 0 :
                    constExpr += x[ k ][ i ][ j ] - x[ 4 ][ i ][ j ]
                    constExpr += ship[ 4 ][ i ][ j ] - new_cargoes[ k ][ i ][ j ]
                else:
                    constExpr += x[ k ][ i ][ j ] - x[ k-1 ][ i ][ j ]
                    constExpr += ship[ k-1 ][ i ][ j ] - new_cargoes[ k ][ i ][ j ]
            myModel.addConstr( lhs = constExpr , sense = GRB.EQUAL , rhs = 0 , name = "cargoes_needed_to_be_car

# create constaint that ensures number of cargoes shipped on day t from i to j is less or equal to
# the availble cargoes
for k in range(no_days):
    for i in range(no_airports):
        for j in range(no_airports):
            constExpr = LinExpr()
            if i != j :
                constExpr += x[ k ][ i ][ j ] - ship[ k ][ i ][ j ]
            myModel.addConstr( lhs = constExpr , sense = GRB.GREATER_EQUAL , rhs = 0 , name = "ship/availble_ai

#register objective and constraints into model
myModel.update()
```

In [6]:
```
#check the status of the model:
curStatus = myModel.status
if curStatus in (GRB.Status.INF_OR_UNBD, GRB.Status.INFEASIBLE, GRB.Status.UNBOUNDED):
    print("Could not find the optimal solution")
    exit(1)

# write model in a file, find optimal solution, print optimal objective value and solution
myModel.write( filename = "homework9.lp" )
myModel.optimize()
print( "optimal objective value " + str( myModel.ObjVal ) )
print( "optimal solution" )
allVars = myModel.getVars()
#for curVar in allVars:
#    print( curVar.varName + " : " + str( curVar.x ) )
```

```
Gurobi Optimizer version 9.5.2 build v9.5.2rc0 (mac64[rosetta2])
Thread count: 8 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 110 rows, 135 columns and 375 nonzeros
Model fingerprint: 0xbd81c98e
Variable types: 0 continuous, 135 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [3e+00, 3e+01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+01, 1e+03]
Presolve removed 60 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 2.172500e+04, 37 iterations, 0.00 seconds (0.00 work units)

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

*    0     0               0    21725.000000 21725.0000  0.00%     -    0s

Explored 1 nodes (37 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 8 (of 8 available processors)

Solution count 1: 21725

Optimal solution found (tolerance 1.00e-04)
Best objective 2.172500000000e+04, best bound 2.172500000000e+04, gap 0.0000%
optimal objective value 21725.0
optimal solution
```

In [7]:

```
print( "optimal objective value ", str( myModel.ObjVal ), '\n' )
myModel.printAttr("X")
```

B Gurobi Solution

```
optimal objective value  21725.0


    Variable            X
-------------------------
repositioning_day1_from2_to1          275
repositioning_day1_from2_to2           20
repositioning_day1_from2_to3          165
repositioning_day2_from2_to1           85
repositioning_day2_from2_to2          220
repositioning_day2_from2_to3          265
repositioning_day3_from2_to1          495
repositioning_day3_from2_to3          165
repositioning_day4_from1_to1          110
repositioning_day4_from2_to1          170
repositioning_day4_from2_to3          180
repositioning_day5_from2_to3          550
shipment_on_day1_from1_to2          15
shipment_on_day1_from1_to3          50
shipment_on_day1_from2_to1          25
shipment_on_day1_from2_to3          25
shipment_on_day1_from3_to1          40
shipment_on_day1_from3_to2         585
shipment_on_day2_from1_to2         290
shipment_on_day2_from1_to3          50
shipment_on_day2_from2_to1          25
shipment_on_day2_from2_to3          25
shipment_on_day2_from3_to1          40
shipment_on_day2_from3_to2         200
shipment_on_day3_from1_to2         100
shipment_on_day3_from1_to3          50
shipment_on_day3_from2_to1          25
shipment_on_day3_from2_to3          25
shipment_on_day3_from3_to1          40
shipment_on_day3_from3_to2         300
shipment_on_day4_from1_to2         400
shipment_on_day4_from1_to3          50
shipment_on_day4_from2_to1          25
shipment_on_day4_from2_to3          25
shipment_on_day4_from3_to1          40
shipment_on_day4_from3_to2         200
shipment_on_day5_from1_to2         295
shipment_on_day5_from1_to3          50
shipment_on_day5_from2_to1          25
shipment_on_day5_from2_to3          25
shipment_on_day5_from3_to1          40
shipment_on_day5_from3_to2         215
cargoes_needed_to_be_shipped_day1_from1_to2          105
cargoes_needed_to_be_shipped_day1_from1_to3           50
cargoes_needed_to_be_shipped_day1_from2_to1           25
cargoes_needed_to_be_shipped_day1_from2_to3           25
cargoes_needed_to_be_shipped_day1_from3_to1           40
cargoes_needed_to_be_shipped_day1_from3_to2          585
cargoes_needed_to_be_shipped_day2_from1_to2          290
cargoes_needed_to_be_shipped_day2_from1_to3           50
cargoes_needed_to_be_shipped_day2_from2_to1           25
cargoes_needed_to_be_shipped_day2_from2_to3           25
cargoes_needed_to_be_shipped_day2_from3_to1           40
cargoes_needed_to_be_shipped_day2_from3_to2          200
cargoes_needed_to_be_shipped_day3_from1_to2          100
cargoes_needed_to_be_shipped_day3_from1_to3           50
cargoes_needed_to_be_shipped_day3_from2_to1           25
cargoes_needed_to_be_shipped_day3_from2_to3           25
cargoes_needed_to_be_shipped_day3_from3_to1           40
cargoes_needed_to_be_shipped_day3_from3_to2          300
cargoes_needed_to_be_shipped_day4_from1_to2          400
cargoes_needed_to_be_shipped_day4_from1_to3           50
cargoes_needed_to_be_shipped_day4_from2_to1           25
cargoes_needed_to_be_shipped_day4_from2_to3           25
cargoes_needed_to_be_shipped_day4_from3_to1           40
cargoes_needed_to_be_shipped_day4_from3_to2          200
cargoes_needed_to_be_shipped_day5_from1_to2          300
cargoes_needed_to_be_shipped_day5_from1_to3           50
cargoes_needed_to_be_shipped_day5_from2_to1           25
cargoes_needed_to_be_shipped_day5_from2_to3           25
cargoes_needed_to_be_shipped_day5_from3_to1           40
cargoes_needed_to_be_shipped_day5_from3_to2          400
```

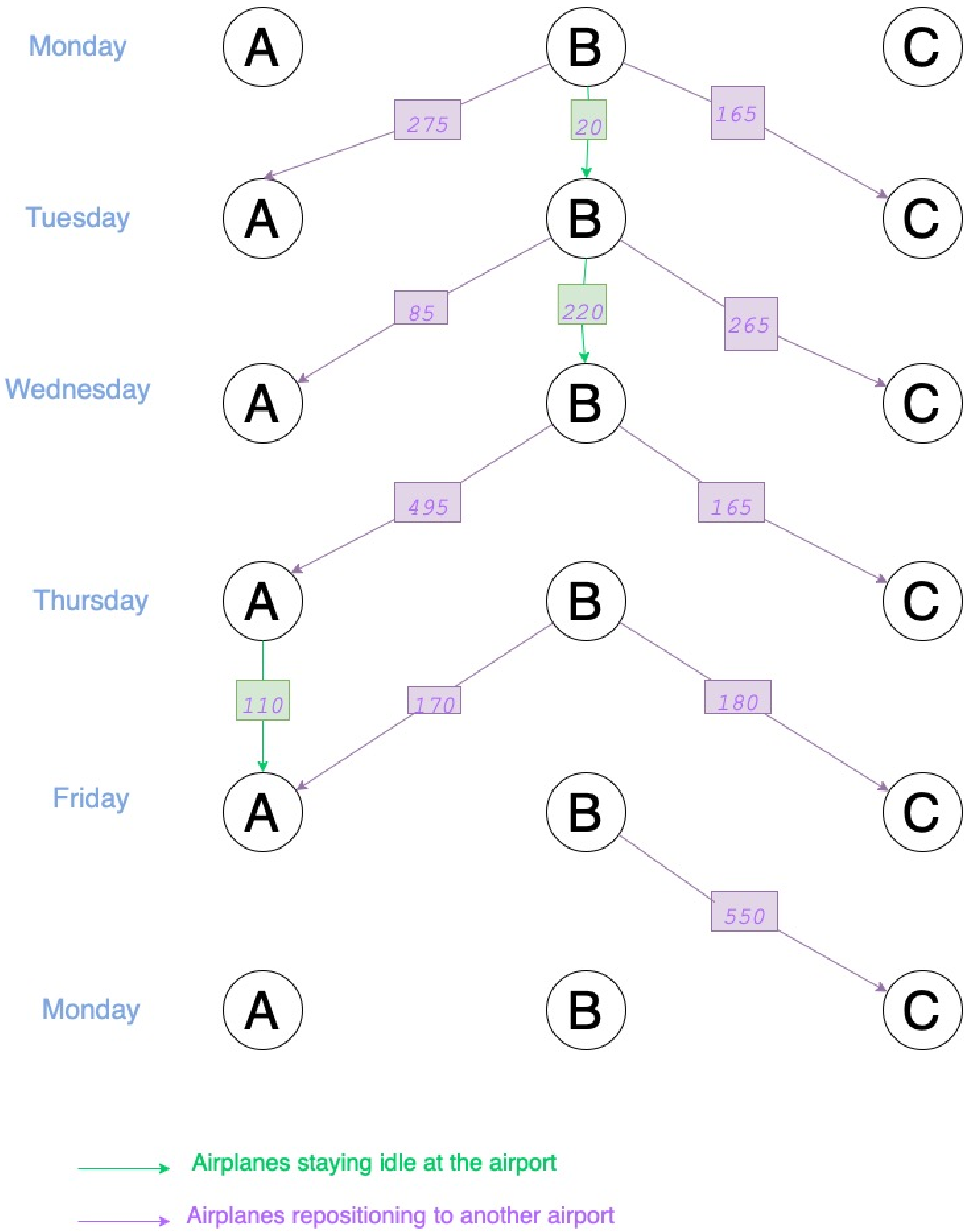
```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

C Empty aircraft flow



D The Proposed Fleet Schedule

Total Cost of the suggested schedule: $210,725,000.00
Number of aircraft available at the beginning of day t:

t	Monday	Tuesday	Wednesday	Thursday	Friday
A	65	340	150	560	345
B	510	620	710	400	600
C	625	240	340	240	255

Number of aircraft used to ship cargo from origin airport to destination airport on day t:

Origin-Destination of the shipment	Monday	Tuesday	Wednesday	Thursday	Friday
A – B	15	290	100	400	295
A – C	50	50	50	50	50
B – A	25	25	25	25	25
B – C	25	25	25	25	25
C – A	40	40	40	40	40
C – B	585	200	300	200	215

Number of aircraft repositioned from origin airport to destination airport on day t:

Origin-Destination of the aircraft reposition	Monday	Tuesday	Wednesday	Thursday	Friday
A – B	0	0	0	0	0
A – C	0	0	0	0	0
B – A	275	85	495	170	0
B – C	175	265	165	180	550
C – A	0	0	0	0	0
C – B	0	0	0	0	0

Number of aircraft staying on the ground in the airport on day t:

	Monday	Tuesday	Wednesday	Thursday	Friday
A	0	0	0	110	0
B	20	220	0	0	0
C	0	0	0	0	0

Number of cargo needed to be shipped at the beginning of day t from origin airport to destination airport on day t:

Origin-Destination of the total cargo needed to be shipped	Monday	Tuesday	Wednesday	Thursday	Friday
A – B	105	290	100	400	300
A – C	50	50	50	50	50
B – A	25	25	25	25	25
B – C	25	25	25	25	25
C – A	40	40	40	40	40
C – B	585	200	300	200	400