**Final Project Report**

**Kamrun Naher Sumi**

**Aircraft Landing**

Kamrun Naher Sumi,

*Department of Data Science and Business Analytics, Wayne State University, MI, USA*

**Problem Statement:**

The goal of this research is to find the airplane landing schedule in a given airport. The algorithm must compute the landing time which satisfies the predetermined separation time window between the landing of a plane and the landing of all the successive planes. Every plane has a predefined target landing time. If the plane lands before or after the target landing time, a penalty cost is imposed. The goal is to minimize this penalty cost. Due to the scope of this work, this analysis only considers the static case and a single runway. For simplicity, static case deals with idealistic situation, i.e. the appearance time of the flights is known even though in reality appearance time is uncertain. A mixed integer linear programming model is utilized to solve this aircraft landing scheduling problem.

**Literature Review:**

Air traffic has drawn immense attention to researchers over the past few decades due to its significant growth. For example, Atlanta, USA airport alone handles more than 80 million aircraft landing and take-off per year. Apart from the increase in air traffic, the limitation of the runway becomes a problem during the airport operation. The busiest airport in the world London Heathrow has only two runways. If the number of approaching flights exceeds the airport capacity, some of these aircrafts cannot land at perfect landing time. Fuel cost increases if the aircrafts hover around the airport or fly at slower or faster than its economical speed. Due to the landing time delay, some passengers may even miss their connecting flights. Therefore, solving aircraft landing problem (ALP) has become very important to solve where each aircraft needs to be assigned an optimal landing time and runway so that the total cost is minimized. Surprisingly, There is not much work done in this field. This ALP is obviously a difficult problem and it can be viewed as a job machine scheduling problem where the program needs to schedule the release times and sequence dependent processing time. The job machine scheduling problem is called a NP-hard (see Beasley et al. (2000)). Over the few decades, exact algorithms and heuristic

algorithms was utilized for the ALP. The exact algorithm is able to provide the optimal solution within reasonable time when the no of aircraft is up to 50. One of the disadvantage of this exact algorithm is that the simulation time increases exponentially with the increase of aircraft numbers. Therefore, researchers have developed heuristics methods to solve this problem quickly (Pinol et al. (2003), Jung et al. (2003)).

**Aircraft-Landing-Schedule Exact Method:**

A Mixed Integer Programming model is leveraged for scheduling aircraft landings for static and single runway case.

**Model Formulation:**

In this static and single runway case, all the data is predetermined and does not change over time. The constraints include earliest landing time, target landing time, latest landing time, penalty cost for landing before and after the target time, and finally separation time between landings.

**Data Set:**

The dataset was collected from the following website, <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/>

There are thirteen single runway static case scenarios given in this website. The only difference among them is the variation of no of planes and its associated variables. In this study, we analyse a handful number of situations instead of analysing all the data, where the no of planes are 10, 50, 100, 150, 250, and 500. A python code was written to read the raw data taken from this website and arranged them in the right order in the excel files. The file name of the python code is Read\_Data.ipynb. The mathematical formulation of this study is demonstrated below.

**Parameters:**

I = No of planes (i = 1, … 10)

J = (j = 1, …10)

P = Total number of flights that have to be scheduled.

= Earliest allowable landing time for plane i.

= Latest allowable landing time for plane i.

= Target allowable landing time for plane i.

= The required separation time (>=0) between plane i landing and plane j landing (where plane i lands before plane j)

= The penalty cost (>=0) per unit for landing before the target time for plane i= (1……P)

= The penalty cost (>=0) per unit for landing after the target time for plane i= (1…P)

M = Big number, where M= Maximum ()latest allowable landing time for plane i minus minimum ()earliest allowable landing time for plane i

**Decision variables:**

= The landing time for plane *i*

= How soon plane *i* lands before

= How soon plane *i* lands after

= {1 if plane *i* lands before plane j; otherwise 0}

**Objective function:**

**Constraints:**

1. Either plane i must land before plane j () or plane j must land before plane i ()

*+*

2.Landing time for each plane should be equal or greater than the earliest allowable landing time. This constraint ensures that each plane lands within its time window.

3.Landing time for each plane should be equal or less than the latest allowable landing time. This constraint ensures that each plane lands within its time window.

4 & 5. Ensure that alpha[i] is at least as big as zero and the time difference between T[i] and x[i], and at most the time difference between T[i] and E[i].

6 & 7. Ensure that beta[i] is at least zero and greater than or equal to the time difference between x[i] and T[i] as well as less than or equal to the time difference between L[i] and T[i].

8. Impose the separation time constraint for pair of planes i and j. There are two scenarios exist in this case: one *deltaji* = 0 when i lands before j; another case is *deltaji* = 1 which means j lands before i.

In this study, we consider the case where either of the plane i or j can land first. The reference paper also demonstrates all the necessary constraints for this case. All the constraints mentioned above have one thing in common, i.e. they are valid when i is not equal to j. This means that the plane cannot make pair of itself.

**MIP Model Result Analysis:**

In this study, we analyse a handful number of situations instead of analysing all the data, where the no of planes are 10, 50, 100, 150, 250, and 500.

Table 1. Analysis results of a static case and single runway ALP.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No of plane** | **No of variable** | **No of constraints** | **Optimal solution found?** | **Optimal value at final step** | **Stopping criteria** | **Gap (%)** | **Run time (s)** |
| 10 | 40 | 231 | Yes | 700 | Not applied | 0 | 3 sec |
| 50 | 200 | 5152 | Yes | 1775 | Not applied | 0 | 5 |
| 100 | 400 | 20301 | No | 5274 | 2 hrs 17 min | 25 | > 2hrs 17 min |
| 150 | 600 | 45452 | No | 12310 | 20 mins | 54 | > 20 mins |
| 250 | 1000 | 125752 | No | 16130 | 20 mins | 45 | > 20 mins |
| 500 | 2000 | 501502 | No | 34378 | 21 mins | 49 | > 20 mins |

Table 1 shows the analysis details of a static case single runway ALP for the number of aircrafts 10, 50, 100, 150, 250, and 500. The number of variables increases from 40 to 2000 with the increase of aircraft number from 10 to 500. On the other hand, the rate of increase of number of constraints are much higher compared to the number of variables with the increase of number of aircrafts. For example, the number of constraints for 10 aircrafts is 231 whereas the number of constraints for 500 aircraft is a little over than half million. As shown in the table, optimal solution is only found for plane number 10 and 50. For more than 50 aircrafts, the convergence rate is extremely slow and optimal solution was not found within the time specified in this table. For the 100 aircrafts, the simulation was run for 2 hours and 17 minutes, yet optimal solution was not found. The gap at final step for 100 aircraft was found to be 25%. Therefore, a stopping criterion of 20 minutes and 20% gap was applied for the rest of the aircrafts. As shown in this table 1, the gap is close to 50% for number of aircrafts more than 100.

Figure 1. optimal value at final step against the no of aircraft.

Figure 1 shows the optimal solution at final step against the number of aircrafts. As shown in this figure, the optimal solution increases with the increase of number of aircrafts since penalty cost is higher for higher number of aircrafts. The trend between aircraft number 250 and 500 seems linear in the above graph because there is no data point between them. If there were more data points between them, the line might have been nonlinear between these two points. This graph shows that more runways are mandatory in order to reduce the cost when number of landing aircrafts are high.

|  |  |
| --- | --- |
|  |  |
|  |  |
| Figure 2. convergence rate for the number of aircrafts (a) 100, (b) 150, (c) 250, and (d) 500. | |

Table 1 suggests us to investigate the convergence rate, i.e. change of optimal solution value over time, for large number of aircraft where optimal solution was found within 20 minutes. Figure 2 shows the convergence rate for the aircraft number 100, 150, 250, and 500. In these graphs, the x axis represents the simulation time and y axis represent the objective function value. For all the cases, it is observed that the objective function value remains either constant or significantly low which can be assumed as constant. This means that even if the simulation is run for longer time, it is highly unlikely to get the optimal solution with 0% gap at reasonable timeframe. For example, the simulation was run for 2 hours and 17 minutes for the 100 aircrafts, but the objective function value was not changing over time. This proves that it is extremely difficult to get exact solution for the aircraft number greater than 50 which matched with the research findings in the research articles. Therefore, a different approach must be taken, i.e. heuristic approach will be a better method to solve this problem faster with the compromise of solution accuracy.

**References:**

[1] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abranmson, Scheduling Aircraft Landings-the static case, Transportation Science, 34(2), 180-197, 2000.

[3] H. Pinol, and J. E. Beasley, Scatter Search and Bionomic Algorithms for the Aircraft Landing Problem, to appear in the European Journal of Operational Research, Currently available from

http://people.brunel.ac.uk/ mastjjb/jeb/jeb.html, 2004.

[4] G. Jung, and M. Laguna, Time segmenting heuristic for an aircraft landing problem, Working paper, Leeds School of Business, University of Colorado, Boulder, CO 80309-0419, USA.

Submitted for publication. Currently available from <http://leedsfaculty>.colorado.edu/laguna/articles/tsh.html, March, 2003.