U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

The Aircraft Landing Problem for a Single Runway

MASTER'S DEGREE IN ENGINEERING AND DATA SCIENCE

BY IGOR DINIZ, INGRID DINIZ AND JOSÉ SANTOS

14-01-2024

An Optimistic Approach for the Aircraft Landing Problem for a Single Runway

Abstract: Abstract: This research addresses aircraft landing scheduling using a dataset from seminal papers in Transportation Science. It employs Mixed Integer Programming (MIP) and Constraint Programming (CP) to minimize penalty costs for landing time deviations. The models handle overlapping time windows and separation constraints efficiently through pre-processing. Computational experiments, conducted on a personal computer, demonstrate that the CP model exhibits a 67.87% average improvement over MIP, with faster execution times. However, CP shows higher memory usage, particularly with larger datasets. Providing insights into the trade-offs between MIP and CP for aircraft landing scheduling.

Keywords: Aircraft Landing problem, Mixed Integer Programming, Constraint Programming.

1. INTRODUCTION

Efficient aircraft landing scheduling is a critical aspect of optimizing airport operations and ensuring passenger safety. This research delves into the complexities of this task, drawing inspiration from two influential papers published in Transportation Science (2000) and the Journal of the Operational Research Society (2004). The dataset, comprising 13 files, serves as the foundation for exploring scheduling strategies, with a particular focus on handling static cases. The mathematical modeling approach embraces both Mixed Integer Programming (MIP) and Constraint Programming (CP) logic, aiming to minimize penalty costs associated with deviations from target landing times. This introduction sets the stage for a comprehensive investigation into the intricate challenges of aircraft landing scheduling and the comparative effectiveness of MIP and CP models in addressing these challenges.

2. PROBLEM DESCRIPTION

In the realm of air traffic control (ATC) at airports, the challenge of scheduling aircraft landings becomes a crucial endeavor. As each aircraft enters the radar range, the ATC must strategically determine the landing time, often referred to as the broadcast time. This landing time is constrained by a specific time window, dictated by the earliest and latest times unique to each plane. The earliest time signifies when the aircraft can land at its maximum airspeed, while the latest time represents the most fuel-efficient airspeed after circling for the maximum allowable duration.

Introducing an additional layer of complexity, each aircraft possesses a preferred cruise speed, and if required to land at this speed, is assigned a target time. Deviations from this target time, whether through adjustments in speed, holding patterns, or accelerations, incur costs that escalate with the disparity between the assigned and target landing times. Furthermore, the temporal gap between the landing of one aircraft and the next, known as separation time, must exceed a predetermined minimum. In navigating these multifaceted considerations, the aim is to optimize landing schedules while minimizing costs associated with deviations and ensuring safe separations between successive landings.

3. DATASET

3.1 Source

The data files originate from two seminal papers: "*Scheduling Aircraft Landings - The Static Case*" published in Transportation Science (2000) and

"*Displacement Problem and Dynamically Scheduling Aircraft Landings*" published in the Journal of the Operational Research Society (2004).

3.2 Dataset Description

The dataset comprises 13 files, with the first eight files specifically designed for the experiments outlined in the Transportation Science paper. Each file follows a standardized format, including the number of planes (P) and freeze time. For each plane $i(i = 1, \dots, P)$, the data includes appearance time, earliest landing time, target landing time, latest landing time, penalty costs for landing before and after the target time, and separation time requirements between each pair of planes.

4. MATHEMATICAL MODELING

The scheduling of each aircraft is formulated initially as a Mixed Integer Programming (MIP) model and subsequently based on Constraint Programming (CP) logic. Both models determine the landing times of each plane on a single runway.

4.1 Pre-processing used in both models.

Before using the constraints directly in the whole set of planes from an "airland", we implemented some pre-processing to resolve the easiest cases and to improve the speed with which the solver handles the remaining unresolved cases.

To do that, we created the following variables to store each pair combination, according to its characteristics:

$\boldsymbol{U_{i,j}}$ **:** Variable which stores a tuple of pairs of planes $(\boldsymbol{i}, \boldsymbol{j})$ that represents the non-obvious cases where time windows of $\boldsymbol{i}$ and $\boldsymbol{j}$ are overlapped. $(\boldsymbol{i} \neq \boldsymbol{j},\ \boldsymbol{i}, \boldsymbol{j} \in \boldsymbol{P})$.

$\boldsymbol{W_{i,j}}$**:** Variable which stores a tuple of pairs of planes $(\boldsymbol{i}, \boldsymbol{j})$ that represents the obvious cases where the time window of $\boldsymbol{i}$ is completely before the time window $\boldsymbol{j}$, and earliest time of $\boldsymbol{j}$ is greater than latest time $\boldsymbol{i}$ + separation time between $\boldsymbol{i}$ and $\boldsymbol{j}$. $(\boldsymbol{i} \neq \boldsymbol{j}, \boldsymbol{i}, \boldsymbol{j} \in \boldsymbol{P})$**.**

$\boldsymbol{V_{i,j}}$: Variable witch stores a tuple of pairs of planes $(\boldsymbol{i}, \boldsymbol{j})$ that represents the obvious cases when the time window of $\boldsymbol{i}$ is completely before the time window $\boldsymbol{j}$, however earliest time of $\boldsymbol{j}$ IS NOT greater than latest time $\boldsymbol{i}$ + separation time between $\boldsymbol{i}$ and $\boldsymbol{j}$. $(\boldsymbol{i} \neq \boldsymbol{j}, \boldsymbol{i}, \boldsymbol{j} \in \boldsymbol{P})$.

Basically $\boldsymbol{W}$ and $\boldsymbol{V}$ are obvious cases where $\boldsymbol{i}$ aircraft lands before $\boldsymbol{j}$, because the Latest Time of $\boldsymbol{i}$ is less than the Earliest Time of $\boldsymbol{j}$. Thus, these cases are resolved, and the problem should not care about which plane lands before.

For the $\boldsymbol{W}$ pairs of planes, it already has the cases of separation time is solved automatically because the Earliest Time of $\boldsymbol{j}$ is greater than the Latest Time of $\boldsymbol{i}$ + separation time between $\boldsymbol{i}$, $\boldsymbol{j}$. However, the same does not occur for the $\boldsymbol{V}$ cases, so we should include for $\boldsymbol{V}$ the separation time constraint.

Therefore, the model itself only deals with $\boldsymbol{U}$ cases because they are cases where the aircraft intervals allowed to land are overlapping.

4.2 Mixed Integer Programming (MIP)

Let $\boldsymbol{P}$ = The number of planes

Our objective is to minimize the total penalty cost for landing time deviations. The complete model is given by (1)– (12).

Objective Function:

Minimize the total penalty cost for landing time deviations:

$$\boldsymbol{min} \sum_{\boldsymbol{i} \in \boldsymbol{P}} (\boldsymbol{PCb_i} \cdot \boldsymbol{\alpha_i} + \boldsymbol{PCa_i} \cdot \boldsymbol{\beta_i}) \tag{1}$$

Decision Variables:

x_i: Actual landing time for each plane i, where i belongs to set P.
α_i: Difference to the target time from the actual landing time when $x_i \le T_i$ for each plane i.
β_i: Difference to the target time from the actual landing time when $x_i \ge T_i$ *f*or each plane i.
δ_{ij}: Binary variable indicating whether plane i lands before plane j (1 if true, 0 otherwise).

Parameters:

E_i: Earliest time a plane i can land.
L_i: Latest time a plane i can land.
S_{ij} : Minimum separation time between planes i and j.
T_i: *Preferred landing time for plane* $i \in P$.
PCb_i : Penalty cost per time-unit for landing before the target time T_i
PCa_i: Penalty cost per time-unit for landing after the target time T_i.

Constraints:

Landing Time Calculation and difference from the Target Time

- $x_i = T_i - \alpha_i + \beta_i$ (2)
- $\alpha_i \ge T_i - x_i$ (3)
- $\beta_i \ge x_i - T_i$ (4)
- $0 \le \alpha_i \le T_i - E_i$ (5)
- $0 \le \beta_i \le L_i - T_i$ (6)
- $E_i \le x_i \le L_i$ (7)

$\forall\ i\ \in P.$

Decision Binary Variable for Landing Order

- $\delta_{ij} + \delta_{ji} = 1, \forall i, j \in P, i \ne j$ (8)
- $\delta_{ij} = 1, \forall i, j \in W\ \cup\ V, i \ne j$ (9)

Separation Time constraint enforcement

$$x_j \ge x_i + S_{ij} \quad \forall(i,j) \in V \qquad (10)$$

$$x_j \ge x_i + S_{ij} - M\delta_{ji} \forall(i,j) \in U \quad (11)$$

$$x_j \ge x_i + S_{ij} \cdot \delta_{ij} - (L_i - E_j) \cdot \delta_{ji}, \ \forall i, j \in U, i \ne j \qquad (12)$$

Where The equation (11) leads to the equation (12).

Before reporting numerical results obtained with this formulation, we shall first present a novel alternate methodology to solve the ALP. Directly solving the MIP leads to high computational times, as we shall show in Section 5.

4.3 Constraint Programming (CP)

Our objective is to minimize the total cost. The complete model is given by (1)–(8).

Objective Function:

Minimize the total cost:

$$min \sum_{i=1}^{P} cost_i \qquad (1)$$

Decision Variables:

t_i: Actual landing time for each plane $i \in P$.
$cost_i$: Cost associated with the landing time for each plane $i \in P$.
$lb_{i,j}$: Boolean variable indicating whether plane lands before plane i lands before plane j, ($\forall\ i, j \in P$).

Parameters:

E_i: Earliest time a plane i can land.
L_i: Latest time a plane i can land.
S_{ij} : Minimum separation time between planes i and j.
T_i: *Preferred landing time for plane* $i \in P$.

PCb_i : Penalty cost per time-unit for landing before the target time T_i.
PCa_i: Penalty cost per time-unit for landing after the target time T_i.

Constraints:

$$E_i \leq t_i \leq L_i \qquad (2)$$

$$0 \leq cost_i \leq \infty \qquad (3)$$

1. Mutual Exclusivity Constraints:

$$lb_{ij} \oplus lb_{ji}, \forall (i.j) \in P, i \neq j \qquad (4)$$
$$lb_{ij} = 1, \forall i,j \in W \cup V, i \neq j \qquad (5)$$

2. Separation Time Constraint

$$x_j \geq x_i + S_{ij} \; \forall (i,j) \in V, i \neq j \qquad (6)$$

$$lb_{ij} \Rightarrow t_i + S_{ij} \leq t_j \;\; \forall i,j \in U, i \neq j \qquad (7)$$

3. Costs Constraints:

$$cost_i = PCb_i \cdot max\,\{0, T_i - t_i\} + PCa_i \cdot max\,\{0, t_i - T_i\} \;\; \forall i \in P \qquad (8)$$

5. RESULTS AND DISCUSSION

In this section, we report the computational and optimal results of the MIP and the CP formulation approach. Experiments were run on a personal computer under Linux operating system, processor Intel(R) Core (TM) i7-8565U 1.80GHz, 16 GB of RAM.

The models were implemented and solved in Python using the OR-Tools and CP-SAT solver libraries.

5.1 Test Instance

In this project, we utilize test instances derived from the seminal research of J.E. Beasley and collaborators, as outlined in their papers focusing on aircraft landing scheduling. These instances, labeled as airland1 through airland13, serve as a foundational component for assessing the performance of scheduling algorithms within the landscape of airport operations.

The results and analysis presented in this study was primarily derived from File 1 to File 8 because computational costs imposed limitations on the extension of the analysis. To deal with Files 9 to 13 we set up a timeout limit to both solvers to return the best solution found after one minute. Because of this, an analysis of these data files for time performance does not make sense, since both models would return 60 seconds. Although, the analysis of memory was implemented without any additional obstacle for all data files.

5.2 Computational Results

To comprehensively evaluate our computational results, we have organized our analysis into two distinct categories: "Execution Time" (as illustrated in Figures 2 and 3) and "Memory Usage" (depicted in Figures 3 and 4).

After analyzing the computational costs of Mixed-Integer Programming (MIP) and Constraint Programming (CP) models, it's clear that CP has a faster execution time than MIP.

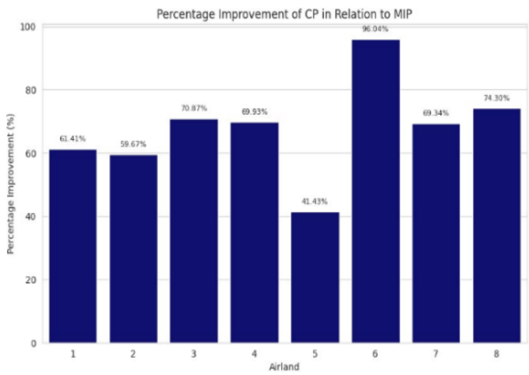

Figure 1 - Improvement Time Execution Percentage of CP to MIP.

Across the first 8 processed files, an average percentage improvement time execution of 67.87% was observed as we can see in Figure 1.

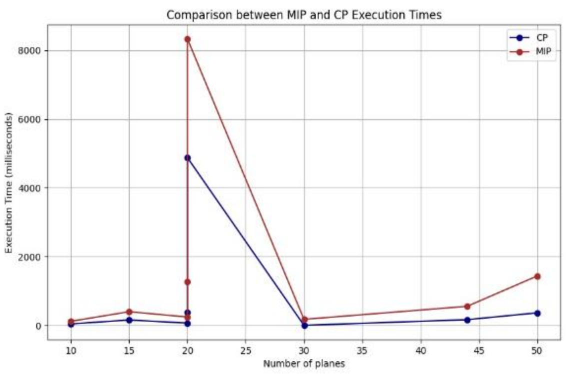

Figure 2 - Comparison between MIP and CP Execution Time.

Yet, this advantage comes with higher memory usage, especially noticeable with larger data sets, as seen in Figure 3.

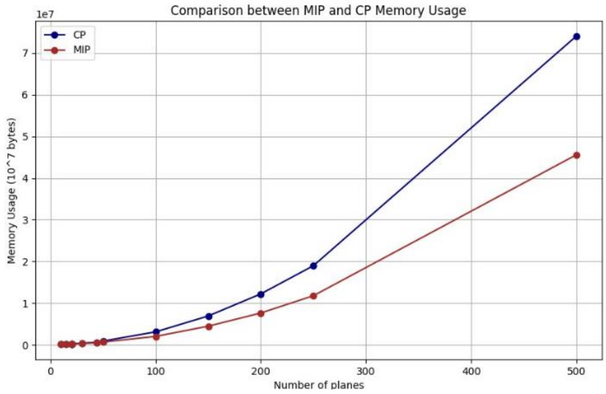

Figure 3 - Comparison Between MIP and CP Memory Usage

When examining the percentage improvement of Constraint Programming (CP) compared to Mixed-Integer Programming (MIP), it becomes apparent that the constraint programming model experiences notable memory losses with the escalation of dataset size, as depicted in Figure 4.

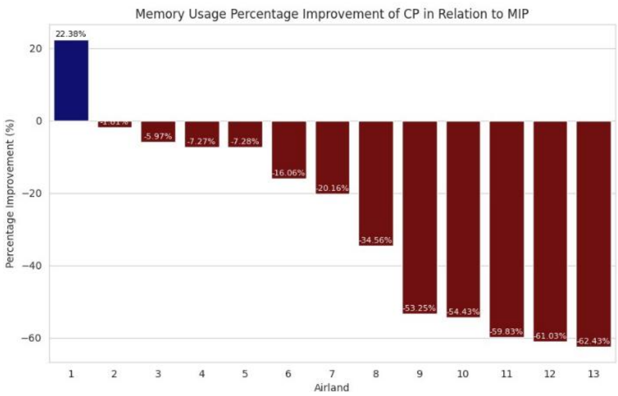

Figure 4 - Memory Usage Percentage Improvement of CP to MIP

5.3 Solution Results

Following the successful implementation of both models, we have achieved optimal solutions for both the Mixed-Integer Programming (MIP) and Constraint Programming (CP) problems. The results have been recorded in Table 1, which not only displays the optimal values achieved but also includes the corresponding number of instances (aircraft) for each file utilized in this study.

	Nº Instances	Optimal Value
Airland 1	10	700
Airland 2	15	1480
Airland 3	20	820
Airland 4	20	2520
Airland 5	20	3100
Airland 6	30	24442
Airland 7	43	1550
Airland 8	50	1950

Table 1 – Results for files 1 to 8

Due to the substantial computational cost observed for many instances (aircraft), as referenced before, a time limit of 1 minute was incorporated into the code development. This limit ensures that the solution obtained within the specified timeframe is returned, even if it is not the optimal solution. It is important to note that these values, although not appended to Table 1, are available in Table 2 for reference.

	Nº Instances	Feasible Value MIP	Feasible Value CP
Airland 9	100	5924.41	5622.40
Airland 10	150	13323.62	13335.83
Airland 11	200	13378.66	13123.62
Airland 12	250	19848.50	19742.71
Airland 13	500	67366.37	64186.56

Table 2 – Results for files 9 to 13

5.4 KPI Results

Finally, at the end of this report we attach a table (Table 3) with a comprehensive set of Key Performance Indicators (KPIs) scrutinized throughout this study. These metrics include Solver Status, MIP Objective Values, MIP Execution Time, MIP Memory Usage, CP Objective Values, CP Execution Time, CP Memory Usage, CP Propagation, and CP Conflicts.

6. CONCLUSION

The comparative analysis between MIP and CP models yields valuable insights.

The CP model, on average, showcases a notable 67.87% improvement over MIP, demonstrating its efficiency in achieving faster execution times. However, it is important to note that this advantage comes at the cost of higher memory usage, particularly evident with larger datasets.

The research underscores the complexity of aircraft landing scheduling, encompassing considerations such as time windows, cruise speeds, and separation times. The findings not only contribute to the understanding of scheduling algorithms but also provide a nuanced exploration of the trade-offs between computational efficiency and memory utilization in the context of optimizing airport operations.

ACKNOWLEDGMENTS

The authors would like to thank professors Luís Gonçalo Rodrigues Reis Figueira and Daniel Augusto Gama de Castro Silva for the classes taught that were fundamental to the development of this project.

REFERENCES

[1] Scheduling Aircraft Landings - The Static Case. (n.d.). [Online]. Available: https://bura.brunel.ac.uk/bitstream/2438/4769/1/Fulltext.pdf

[2] Problem Definition. (n.d.). [Online]. Available: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/airlandinfo.html

[3] Displacement Problem and Dynamically Scheduling Aircraft Landings. (n.d.). [Online]. Available: https://www.jstor.org/stable/4101827

[4] An Optimistic Planning Approach for the Aircraft Landing Problem. (n.d.). [Online]. Available: https://enac.hal.science/hal-02382692/document

COPYRIGHT

ATTACHMENT

Airland	N Planes	Solver Status	MIP Obj. Value	MIP Execution Time	MIP Memory Usage
1	10	OPTIMAL	700.0	174	267187
2	15	OPTIMAL	1480.0	449	221191
3	20	OPTIMAL	820.0	338	255705
4	20	OPTIMAL	2520.0	965	251129
5	20	OPTIMAL	3100.0	5683	250905
6	30	OPTIMAL	24442.0	185	339590
7	44	OPTIMAL	1550.0	545	534658
8	50	OPTIMAL	1950.0	1809	686949
9	100	FEASIBLE	5924.41	122212	2043207
10	150	FEASIBLE	13323.62	124158	4492676
11	200	FEASIBLE	13378.66	126758	7656575
12	250	FEASIBLE	19848.5	129369	11792521
13	500	FEASIBLE	67366.37	151097	45565241

Airland	CP Obj. Value	CP Execution Time	CP Memory Usage	CP Propagations	CP Conflicts
1	700.0	31.5	207400	803	111
2	1480.0	144.07	225201	2467	1362
3	820.0	35.29	270966	1218	94
4	2520.0	305.89	269396	3142	1711
5	3100.0	2331.05	269173	24451	19159
6	24442.0	5.78	394142	0	0
7	1550.0	139.58	642450	2063	734
8	1950.0	356.56	924329	5179	428
9	5622.4	60069.63	3131183	106312	72983
10	13335.83	60091.18	6937838	96761	55974
11	13123.62	60080.75	12237376	63695	42265
12	19742.71	60080.96	18989334	51783	25478
13	64186.56	60101.37	74012450	72579	15131

Table 3 - Key Performance Indicators (KPIs)