Scheduling Aircrafts Landings under Mixed Integer Programming & Constraint Programming - The Static Case

Cátia Teixeira, Sónia Ferreira and Vasco Bartolomeu
Master's degree in Data Science & Engineering, Faculty of Engineering - University of Porto
Analytical Decision Support Systems

January 2023

http://people.brunel.ac.uk/~mastjjb/jeb/orlib/airlandinfo.html

Abstract: This paper studies the problem of Scheduling Aircraft Landings, an important practical problem in today's world, as the industry suffered an overall tremendous growth in the past decades, and in many countries, effective use must be made of the available runway capacity. The problem consists of determining how to land aircraft approaching an airport and involves assigning each aircraft to an appropriate runway, by computing a landing sequence for each runway and scheduling the landing time for each aircraft. The main objective is to achieve effective runway use. For this purpose, the Aircraft Landing Problem is considered. Two techniques will be presented: Mixed Integer Programming and Constraint Programming. Computational results, performed on publicly available problems involving up to 50 aircrafts and up to 2 runways, show the efficiency of each model.

Keywords: Aircraft landing scheduling, Delay minimization, sequence-dependent scheduling, Runway operation, Mixed integer programming, Constraint programming.

1. Introduction

Over the past few decades, air traffic has experienced tremendous growth, as air transport become one of the fundamental modes of transport for personal and business travel, and commercial delivery. However, in 2020 due to the worldwide COVID pandemic (that started that same year), the industry observed a setback in their numbers, with less traffic for passengers and freight traffic their growth has dropped tremendously [1]. In the next year (2021) the numbers rose to show an improvement from 2020, but they are still very far away from the pre-pandemic numbers [2].

As air traffic developed, the limitation of resources, like manpower but especially the limitation of runways became a bottleneck during airport operations. Located in Europe, London Heathrow airport is one of the busiest airports in the world and has only two runways. When the number of approaching flights overpasses the airport capacity, some of these aircraft can't land on their "target" landing time. Resulting in an extra cost mainly on a waste of fuel for each plane flying faster than its most economical speed. Airlines will have to deal also with costs for delays of their fights and unsatisfied customers. Transfer customers can miss their connecting flights. The crew operating the current flight can be needed on another flight, which has now to be rescheduled resulting in another extra cost. Flights that are on land (departing flights) can also be affected by this, as they can also be delayed and not authorized to depart due to the lack of available runways, this will also have an impact on the operations of the destination airport of these flights. Other possible costs are crew overtime payments, crew rescheduling, etc. Back in 2017, it was reported that congestion cost airlines and passengers around 25 billion euros, according to FAA/Nextor estimated [3].

Therefore, even now that the air industry is not growing or even having the same results as in past years it's important to solve the problem of scheduling aircraft landings which is referred to as the Aircraft Landing Problem (ALP) in the literature. ALP consists of the problem of assigning each aircraft an optimal landing time and runway in a way that the cost is minimized. This can be achieved by reaching the maximum efficiency of resources and overcoming the problems observed in the past decades when the increase in air traffic caused a drastic increase in the number of aircraft take-offs and landings, within a given period at a certain airport or runway, that results in an overload issue in terms of airport capacity and delay issue in terms of aircraft scheduling.

Air traffic control (ATC) aims to control air traffic, to prevent collisions and delays. ATC is usually operated by humans and therefore human error can happen. An important part of their responsibility is the planning of airport operations, such as the arrival and departure of aircraft, which is the focus of this paper. Automating this solution, will not only save time but overcome human error.

Aircraft landing scheduling can be understood as giving priority to different aircraft, that need to land at a certain time. This problem becomes more complex, as previously explained, in busy airports with limited runways and with several aircraft trying to land at the same period.

But there are different ways of formulating this problem, depending on the perspective used. For instance, from a point of view of an airline, their main objective would be to minimize the deviation from the "target" landing time, while from an airport management perspective is to maximize the airport capacity usage and therefore minimize their cost in losses. Both objectives are directly related to cost and the final objective is to minimize direct and indirect costs associated with aircraft landing for both airliners and airport managers.

In resume, air transportation has established itself as one of the most important means of transport, which directly implies the increase in air traffic, and therefore the efficient management and scheduling of aircraft take-offs and landings (given the limited resources such as time, budget, etc..) have become a very challenging and complex problem for air traffic controllers.

Therefore, this paper will consider the problem of scheduling aircraft landings, at a given airport, for multiple runway airports. The problem is considering a landing time on a runway for each plane for a given set of planes such that:

- each plane lands on the predefined window.
- separation criteria between the landing of a plane, and the landing of all successive planes, is respected.

Mixed Integer Programming and Constraint programming will be used to address the problem, formulating this from the point of view of airport management. The main goal is to find an optimal landing sequence based on the available runways, number of flights, and expected delays and therefore minimize their cost.

The aircraft scheduling problem has been widely studied in the operation research community, and therefore this research will adopt its foundation from the paper "Scheduling Aircraft Landings—The Static Case" by J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha and D. Abramson [4].

As in their study, is important to refer that through our paper we will typically refer to planes landing, but the models presented can be applied to problems involving just takeoffs or to problems involving a mix of landing and takeoffs on the same runway. Also, we are dealing only with the static case, where we have all the knowledge about the set of planes that are going to land and no information (i.e. planes land, new planes appear, etc) changes.

The organization of the paper is as follows. In section 2, an overview of the problem context is given. Section 3, describes the formulation adopted. Section 4 introduces the models used to approach the Aircraft Landing Problem, followed by a brief discussion about the results obtained in section 5. In section 7, the paper is concluded by reaching its final conclusions followed by recommendations for future research in order to improve this study.

2. Problem Context

Air traffic control will give instructions to each aircraft entering within the range of an airport radar. A landing time and a runway will be assigned to each plane. The landing time must be comprehended between the earliest landing time and the latest landing time.

The earliest landing time corresponds to the time at which the aircraft can land if it flies at its fastest speed, and the latest time corresponds to the maximum landing time achievable considering delaying mechanisms, such as decreasing the speed of the plane, or if the flight plan can be lengthened by circling. Comprehended in this time window there is the target time that corresponds to the time at which the aircraft can land if it flies at its cruise speed. This is considered the preferred landing time (target landing time).

To ensure safety is necessary to ensure that separation distances are respected. Separation distances are converted into separation times using a fixed landing speed that will be different according to the type of aircraft that is landing. After this, a minimal lapse of time between the landing of a plane and the landing of any successive plane needs to be ensured. Separation time holds between a pair of planes landing on the same runway or on different runways.

In resume, the Aircraft Landing Problem (ALP), concerns the scheduling of planes at an airport by assigning to each plane a runway and a landing time that falls within the specific time window and does not violate the safety constraint (separation time) in a way that the cost is minimized. When it's not possible to land the plane at the target landing time, the plane can be landed at any time that falls between the early and the late landing time, but this will incur a cost (penalty cost).

Therefore, the objective function, considered in this study is the minimization of the total cost associated with landing planes at times that differ from the target landing time, for the static case in a multi-runway airport.

3. Problem Formulation

3.1. Based on State of Art

In this section (for conciseness) the multi-runway ALP problem is formulated, to minimize the cost adopted from [4]. The adopted formulation is as follows:

The notation is as follows:

P	number of planes
E_i	the earliest landing time for plane i (i=1, … , P)
L_i	the latest landing time for plane i (i=1, … , P)
T_i	the target (preferred) landing time for plane i (i=1, … , P)
S_{ij}	the required separation time (≥ 0) between plane i landing and plane j landing (where plane i lands before plane j), i=1,...,P; j=1,...,P; i≠j
g_i	*be the penalty cost (≥ 0) per unit of time for landing before the target time T_i for plane i (i=1,...,P)*
h_i	*be the penalty cost (≥ 0) per unit of time for landing after the target time T_i for plane i (i=1,...,P)*
s_{ij}	It's the required separation time (≥ 0) between plane i landing plane j landing (where plane i lands before plane j and they land on different runways)

Note that:

- for S_{ij}, L_i and E_i these values are integer, this is extremely relevant for the constraint programming model.

- s_{ij}, as recommended by the paper [4] it was considered 0.

The decision variables are:

x_i *the landing time for plane i (i=1,...,P)*

α_i *how soon plane i (i=1,...,P) lands before* T_i

β_i *how soon plane i (i=1,...,P) lands after* T_i

δ_{ij} *1 if plane i lands before plane j (i=1,...,P; j=1,...,P; i≠j)*

0 otherwise

Due to security concerns, is extremely important to ensure that the separation constraint between landings is satisfied. Because although sometimes for a certain pair (i, j) of planes it's possible to observe clearly if $\delta_{ij} = 1$ or $\delta_{ji} = 1$. However, knowing the order in which a set of planes should land doesn't mean that the separation constraint is automatically fulfilled.

Therefore, three sets were defined:

U the set of pairs (i,j) of planes for which we are uncertain whether plane i lands before plane j or not

$$U = [(i, j)\ i = 1, \ldots, P;\ j = 1, \ldots, P;\ i \neq j;\ E_j \leq E_i \leq L_j \text{ or } E_j \leq L_i \leq L_j \text{ or } E_i \leq E_j \leq L_i \text{ or } E_i \leq L_j \leq L_i] \quad (1)$$

V the set of pairs (i,j) of planes for which i definitely lands before j (but for which the separation constraint is not automatically satisfied)

$$V = [(i, j)\ L_i < E_j \text{ and } L_i + S_{ij} > E_j\ i = 1, \ldots, P;\ j = 1, \ldots, P;\ i \neq j] \quad (2)$$

W the set of pairs (i,j) of planes for which i definitely lands before j (and for which the separation constraint is automatically satisfied)

$$W = [(i, j)\ L_i < E_j \text{ and } L_i + S_{ij} \leq E_j\ i = 1, \ldots, P;\ j = 1, \ldots, P;\ i \neq j] \quad (3)$$

Note that by introducing these three sets into the formulation, the number of restrictions of the formulation constraints decreased.

Objective Function:

$$min \sum_{i=1}^{p} (g_i \alpha_i + h_i \beta_i) \quad (4)$$

Subject to (constraints):

$$E_i \leq x_i \leq L_i \qquad i = 1, \ldots, P \quad (5)$$

$$\delta_{ij} + \delta_{ji} = 1 \qquad i = 1, \ldots, P;\ j = 1, \ldots, P;\ j > i \quad (6)$$

$$\delta_{ij} = 1 \qquad \forall (i, j) \in W \cup V \quad (7)$$

$$x_j \geq x_i + S_{ij}\delta_{ij} - (L_i - E_j)\delta_{ji} \qquad \forall (i, j) \in U \quad (8)$$

$$\alpha_i \geq T_i - x_i \qquad i = 1, \ldots, P \qquad (9)$$

$$0 \leq \alpha_i \leq T_i - E_i \qquad i = 1, \ldots, P \qquad (10)$$

$$\beta_i \geq x_i - T_i \qquad i = 1, \ldots, P \qquad (11)$$

$$0 \leq \beta_i \leq L_i - T_i \qquad i = 1, \ldots, P \qquad (12)$$

$$x_i = T_i - \alpha_i + \beta_i \qquad i = 1, \ldots, P \qquad (13)$$

$$\sum_{r-1}^{R} y_{ir} = 1 \qquad i = 1, \ldots, P \qquad (14)$$

$$Z_{ij} = Z_{ji} \qquad i = 1, \ldots, P;\ j = 1, \ldots, P;\ j > i \qquad (15)$$

$$Z_{ij} \geq Y_{ir} + Y_{jr} - 1 \qquad i = 1, \ldots, P;\ j = 1, \ldots, P;\ j > i;\ r = 1, \ldots, R \qquad (16)$$

$$x_j \geq x_i + S_{ij} z_{ij} + s_{ij}(1 - z_{ij}) \qquad \forall (i, j) \in V \qquad (17)$$

$$x_j \geq x_i + S_{ij} z_{ij} + s_{ij}(1 - z_{ij}) - (L_i + max(S_{ij}, s_{ij}) - E_j)\delta_{ji} \qquad \forall (i, j) \in U \qquad (18)$$

The objective function (4) will minimize the total costs of deviation from the target time (T_i).
Constraint (5) ensure that the scheduled landing time for each aircraft lies within its time window. Constraint (6) ensure that aircraft i will land before aircraft j (δ_{ij} = 1) or aircraft j lands before aircraft i (δ_{ji} = 1). Constraint (7) represent the pairs of planes (i, j) for which i lands before j and for which is not always guaranteed that the separation constraint is always satisfied (union of set W with V). Constraint (8) ensure a separation constraint for the pairs of planes in U. Constraint (9) and (10) ensure that α_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. Constraints (11) and (12) are similar equations for β_i. Constraint (13) relate the landing time (x_i) to the time plane i lands before (α_i), or after (β_i), target (T_i). Constraint (14) ensure that each plane lands on exactly one runway whereas constraint (15) are symmetry constraints (meaning that, if i and j land on the same runway so do j and i). Constraint (16) ensure that, if there is any runway r on which plane i and j are both landed (i.e. $y_{ir} = y_{jr} = 1$), then we force z_{ij} to be 1 (i and j land on the same runway). If $z_{ij} = 0$, then constraint (16) become $0 \geq y_{ir} + y_{jr} - 1$, ensuring that planes i and j cannot land on the same runway. Constraints (17) and (18) fulfil the requirement that the separation time is S_{ij} for planes landing on the same runway but s_{ij} for planes landing on different runways can be easily dealt with it, for set V and U respectively.
The formulation described above is then used in the programming of the models of Mixed Integer Programming and Constraint Programming.

3.2. Alternative formulation

Using the same notation as in 33.1, an alternative formulation is also proposed, aiming to improve runtime performance.

The decision variables are:

x_i *the landing time for plane i (i=1,...,P)*

γ_i *runway on which plane (i=1,...,P) lands*

ε_i *1 if plane i (i=1,...,P) lands earlier than target time*

0 otherwise

Objective Function:

$$min \sum_{i=1}^{p} (T_i - x_i)\, g_i\, \varepsilon_i + (x_i - T_i)\,(1 - \varepsilon_i) h_i \qquad (19)$$

Subject to (constraints):

$$E_i \leq x_i \leq L_i \qquad i = 1, \dots, P \qquad (20)$$

$$0 \leq \gamma i \leq R - 1 \qquad i = 1, \dots, P \qquad (21)$$

$$x_i < T_i \quad \Rightarrow \quad \varepsilon_1 = 1 \qquad i = 1, \dots, P \qquad (22)$$

$$x_i \geq T_i \quad \Rightarrow \quad \varepsilon_1 = 0 \qquad i = 1, \dots, P \qquad (23)$$

$$\gamma_i = \gamma_j \quad \Rightarrow \quad x_j \geq x_i + S_{ij} \qquad i = 1, \dots, P;\ j = 1, \dots, P;\ j > i \qquad (24)$$

In this formulation, the use of α_i and β_i is eliminated by introducing a binary variable that assumes the value of 1 when the plane arrives before the target time and 0 otherwise.

The objective function (19) minimizes the total cost of deviation from target time (T_i), however, its formulation was adapted to use the difference between T_i and x_i to calculate the cost.

Constraint (20) ensures that the scheduled landing time for each aircraft lies within its time window (the same as constraint 2). Constraint (21) ensures a given plane lands on existent runways (R). Constraint (22) ensures that if a plane lands before the target time, then ε_1 is 1, while constraint (23) ensures that if a plane lands on or after the target time, then ε_1 is 0.

Constraint (24) ensures if two planes land on the same runway, then their separation times S_{ij} must be respected.

The formulation described above is used in the programming of the optimized model of Constraint Programming.

4. The Models

The adopted and proposed formulation in the previous section is implemented and solved via Docplex, the Python API of CPLEX solver. The instances are generated from the data obtained from OR-Library [5].

4.1. Mixed Integer Programming (MIP) Model

The main considerations involved in the implementation of the MIP model for the Aircraft Landing problem are modelling the model as declared in section 3 for:

- Notation;
- Decision variables;
- The sets U, V and W: this allows the creation of time sets and therefore reduces the number of restrictions to declare in constraints that make use of the time sets;
- Objective function.

This model adopted the formulation in section 3.1. “Based on State of Art”.

4.2. Constraint Programming (CP) Model

4.2.1. CP Model

The first implementation of the CP model followed the same logic and formulation as the MIP model, just oriented to Constraint programming.

4.2.2. CP Model Optimized

Based on orientations taken from [6] and in order to improve performance times and results of the first attempt done implementing the CP algorithm, the bellow enhancements were done:

- After reading the instance files, this information will be ordered by the target landing time of each aircraft. To allow a more optimizable reading of the information during the model formulation;
- The number of workers was predefined to 6 for a faster model;
- Symmetry breaking: according to [6] symmetry breaking can be used in order to get better computation times;
- Replace *sum* from python by *mdl.sum*, this should allow an increase in operating performance.
- Binary variables are now created with a binary_var_list;
- Reused the number of decision variables;
- Predefined the Search type for Restart;

- Constraints are added in batches instead of single-step insertion (improves runtime by around 50%).

This model adopted the formulation in section 3.2. “Alternative Formulation”.

5. Computational Results

The Mixed Integer Programming (MIP) and Constraint Programming (CP) models are implemented and solved via Docplex, the Python API of the CPLEX solver.

As previously referred, the formulation adopted for the multi-runway ALP is from [4], however, due to time restrictions for the preparation, design, interpretation and execution of the computational results that are the end result of this study, it wasn’t possible to include the complete and more strengthened multiple runway formulation (model) proposed by the paper adopted, in the section “5.3 - Complete formulation”. More details about the not included constraints can be found in section (7). Additionally, in order to reinforce the models in this study constraints (8) was introduced in this study to ensure a separation constraint for the pairs of planes for which uncertainty exists if either plane i lands before j or not.

For coherence, computational tests run on an Asus-Windows 11 Laptop, from 2019, with a processor Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz with 16GB of RAM. Using instances publicly available from OR-Library [5], computational tests were performed, depending on the model, up to 8 instances (tests), involving from 10 to 50 aircrafts and up to 2 runways.

Each instance was done by increasing the number of runways up to two. As this happen was possible to observe that the optimal solution (minimum cost) dropped in every instance, which seems to indicate that as more runways become available more planes seem to be able to land on the target landing time. Thus, is possible to hope and envision that, if given more time, by continuing to increase the number of runways for each instance, eventually the optimal solution would reach the value of zero.

Table 1, table 2 and table 3, show the overall computational results for all instances of the MIP, and CP models, respectively. In these tables for each instance, the number of planes is provided, followed by the number of runways, the resolution time of each model and the optimal solution (minimum cost) reached.

Table 1 – Computational Results of the MIP Model

	Original data	Number of planes	Number of runways	MIP-time	MIP-cost
0	airland1.txt	10	1	0.078	700.0
1	airland1.txt	10	2	0.047	90.0
2	airland2.txt	15	1	0.094	1480.0
3	airland2.txt	15	2	0.109	210.0
4	airland3.txt	20	1	0.078	820.0
5	airland3.txt	20	2	0.094	60.0
6	airland4.txt	20	1	0.547	2520.0
7	airland4.txt	20	2	0.578	640.0
8	airland5.txt	20	1	2.688	3100.0
9	airland5.txt	20	2	0.484	650.0
10	airland6.txt	30	1	0.063	7328.0
11	airland6.txt	30	2	0.109	394.0
12	airland7.txt	44	1	0.031	278.0
13	airland7.txt	44	2	0.016	0.0
14	airland8.txt	50	1	0.813	1950.0
15	airland8.txt	50	2	0.907	135.0

Table 2 - Computational Results of the CP Model

	Original data	Number of planes	Number of runways	CP-time	CP-cost
0	airland1.txt	10	1	0.28	700
1	airland1.txt	10	2	0.09	90
2	airland2.txt	15	1	3.11	1480
3	airland2.txt	15	2	0.49	210
4	airland3.txt	20	1	2.88	820
5	airland3.txt	20	2	0.79	60
6	airland4.txt	20	1	208.92	2520
7	airland5.txt	20	1	245.78	3100
8	airland5.txt	20	2	727.64	650
9	airland6.txt	30	1	0.27	7328
10	airland6.txt	30	2	0.83	394
11	airland7.txt	44	1	3.22	278
12	airland7.txt	44	2	0.08	0

Table 3 - Computational Results of the CP Optimized Model

	Original data	Number of planes	Number of runways	CP_opt-time	CP_opt-cost
0	airland1.txt	10	1	0.073	700
1	airland1.txt	10	2	0.357	90
2	airland2.txt	15	1	0.192	1500
3	airland2.txt	15	2	2.830	210
4	airland3.txt	20	1	0.578	1380
5	airland3.txt	20	2	38.107	60
6	airland4.txt	20	1	0.198	2520
7	airland5.txt	20	1	0.250	5420
8	airland5.txt	20	2	1224.230	770
9	airland6.txt	30	1	0.030	24442
10	airland6.txt	30	2	0.320	636
11	airland7.txt	44	1	0.036	1550

For 8 instances, involving up to two runways, the Mixed Integer Programing Model gave a median computation time of 0,101 seconds. The Constraint Programming model gave a median computation time of 0,83 seconds when tested with up to 7 instances and up to two runways[1]. Nevertheless, even with fewer instances executed for the CP model is possible to say that Constraint Programming is the model less efficient in computation terms, in terms of the optimal solution, as the moment as both models use the same formulation as a foundation (section 3.1) the results are identical.

Due to the low efficiency demonstrated in terms of computational times for the CP model, another programming model was done for CP (CP Optimized - described in section 4.2.2), as result when this model was tested up to 7 instances and up two runways[2], the model gave a median computation time of 0,28. Therefore, it's visible that CP Optimized model has a better computational time than the previous model.

In the following tables (table 4 and table5) we can see that the overall costs obtained for the first 3 problems are basically the same, apart from the optimized Constraint Programming model we implemented which has a heuristic approach, property that is supported by the values obtained in the paper we based the project in. Our models were not as efficient as we expected and had high computational costs to solve problems following number 9 and due to the lack of time to present the project the final results aren't displayed.

Until problem 8 our MIP model was able to solve all problems for single and multi-runway, having the same optimal solutions as the [4], apart from problem 6 where we had even better optimal solutions.

[1] Except for instance 4 and runway 2, which was not able to run within 24 hours of execution.
[2] except for instance 4 and 7 and runway 2 which was not able to run within 24 hours of execution

The first Constraint programming model we implemented was based on the MIP structure, so we expected it not to be as efficient as the MIP because this model isn't optimized to work with the mixed-integer programming structure. Having this being said, we weren't able to get results due to computational cost efficiency for problem 4 with 2 runways and problem 8 because of the higher number of planes involved.

The optimized Constraint programming model was designed according to the desired CP model structure, so we hope this model to get the expected results but have a higher computational cost efficiency which is supported in the next table. Although this model is more efficient for the first few problems where the number of constraints was low due to the lower number of planes, once we start increasing the number of constraints the results exponentially got worse so this model has its drawbacks due to the number of constraints.

Table 4 - Comparation for Computational Time of all models vs adopted paper [4]

File	Runways	MIP	CP	CP Optimized	Optimal Paper [4]
1	1	0,078	0,28	0,073	0,4
1	2	0,047	0,09	0,357	0,6
2	1	0,094	3,11	0,192	5,2
2	2	0,109	0,49	2,83	1,8
3	1	0,078	2,88	0,578	2,7
3	2	0,094	0,79	38,107	3,8
4	1	0,547	208,92	0,198	220,4
4	2	0,578	over24h	over24h	1919,9
5	1	2,688	245,78	0,25	922
5	2	0,484	727,64	1224,23	11510,4
6	1	0,063	0,27	0,03	33,1
6	2	0,109	0,83	0,32	1568,1
7	1	0,031	3,22	0,036	10,6
7	2	0,016	0,08	over24h	None
8	1	0,813	over24h	over24h	111,9
8	2	0,907	over24h	over24h	3450,6

Table 5 – Comparation for Cost Times all models vs adopted paper [4]

File	Run	MIP	CP	Optimized CP	Optimal paper [4]	Heuristic
1	1	700	700	700	700	700
1	2	90	90	90	90	90
2	1	1480	1480	1500	1480	1500
2	2	210	210	210	210	210
3	1	820	820	1380	820	1380
3	2	60	60	60	60	60
4	1	2520	2520	2520	2520	2520
4	2	640	None	None	640	640
5	1	3100	3100	5420	3100	5420
5	2	650	650	770	650	1070
6	1	7328	7328	24442	24442	24442
6	2	394	394	636	554	882
7	1	278	278	1550	1550	1550
7	2	0	0	None	0	0
8	1	1950	None	None	1950	2690
8	2	135	None	None	135	255

At first glance at the table we can confidently say that the most efficient model overall is the MIP model because of its consistent considerable low computational times across all problems.

The same can't be said for both Constraint programming models. The first implementation of the CP model with the MIP structure performed poorly as expected because this structure of model is not adequate as was already mentioned before. This model overall performed worse in every single problem having for problem 4 for the multi-runway case and problem 8 forward not being able to solve the problem in a reasonable time.

For the optimized Constraint programming model we saw a different behaviour. For the smaller size problems it was the most efficient model by far in terms of computational cost but when we start increasing the size of the problem its efficiency decreases heavily.

Overall, we could see the strengths of each different model, being the MIP more universal for this kind of problem across different sizes and the CP

Alternatively, the detailed computational results for each instance of each model can be found in the notebooks or the excel files provided in the attachment to the paper. In these tables for each instance (test/file), it's provided, how soon a plane lands before the target line; how soon a plane lands after the target line; landing time for a plane; where the plane lands - binary variable.

6. Conclusions

An approach to scheduling the landing of aircraft based on mixed integer programming (MIP) and constraint programming (CP) was introduced. The proposed formulation considers the multiple runway problem for the static case. Computational results were presented for several test problems involving up to 50 planes and two runways.

As the formulation of the MIP and CP model advanced into the current final formulation, was possible to observe a shift in the results as the formulation gained more strength by introducing more constraints that were recommended in the adopted paper [4].

Computational results indicate that as more runways were introduced into each instance, the close the models got to the optimal solution value (zero cost), indicating that this way all aircrafts could land on their target time.

By modelling the CP model accordingly to the initial formulation of the MIP model, it was possible to observe that the CP model execution was not efficient. Thus, a second and third optimization for the CP model was attempted to improve the time execution of the model. The third optimization was the most promising one in terms of execution time for the CP model, although the MIP model still gets better results.

When comparing the results of our models for the optimal value (cost) got for each instance/runway against the results available in the adopted paper [4], it's possible to observe that the result obtained by our developed algorithms are very similar.

It's possible to envision that the proposed models could improve the management operations of an airport, as described in the introduction, by automating the solution and reducing the cost to a minimum or even to zero if enough runways were available.

7. Future Research

Due to time restrictions, influenced by the excessive time of execution of certain instances and/or models, there were certain developments improvements and tests that were not possible to conduct at the time of the conclusion of this paper. Therefore, on future research and before any other consideration our recommendation is:

- Run all instances available publicly available from OR-Library [5]

Conduct tests on all instances, by increasing the number of runways until the optimal solution drops to zero (this should indicate that all planes are able to land on the target landing time if enough runways are available).

- Strengthening the present static case, multi-runway formulation for ALP

  The multiple runway formulation, of the static case of ALP present in this paper, was adopted from [4], and has been implemented in its entirety. Additionally, constraint (8) was introduced in this formulation to reinforce the formulation's strength.

  However, in the adopted paper, they propose to strengthen their original formulation for the multi-runway with constraints (40)-(43), (45), (47) and (48), which they believe to have computational benefits. We believe that the current models would benefit from these improvements.

  Constraints (40) to (43), will deal with tightening of the delta setting for landing between planes i and j on the same runway or different runways, and the associated gap between the target times of these planes. Constraints (45) and (47) will ensure the improvement in the order landing, turning this more optimal. Constraints (48) will develop a lower bound on the number of pairs of planes that can land on the same runway.

Furthermore, as a result of the work conducted in this paper, the following future research is suggested:

- Investigation of the dynamic case of ALP, for multi-runaway

  The research in this paper focused on the static case of the ALP. This can be an interesting way of investigating and planning the airport's runway capacity in a strategic planning stage. Nevertheless, in the day-to-day operations, the information on the landings, such as the earliest, target, and latest time might change. Therefore, a dynamic approach seems a crucial next step.

- Investigation of the single runway formulation of ALP for the static case and dynamic case

  Most of the busiest international airports have at least two runways but can have more. Therefore, for this initial study efforts were made to approach this formulation that can affect a wide range of airports and deal with different types of problems (i.e. different types of planes, overlapping landings, runway unavailability, …). Nevertheless, the single runway formulation should also be investigated since it will cover a range of airports not covered in this paper, allowing improvement in the airport's runway capacity for the strategic planning stage in the static case and the day-to-day operations in the case of the dynamic case.

References

[1] International Civil Aviation Organization, "Presentation of 2020 Air Transport Statistical Results," [Online]. Available: https://www.icao.int/annual-report-2020/Documents/ARC_2020_Air%20Transport%20Statistics_final_sched.pdf. [Accessed 01 2023].

[2] International Civil Aviation Organization, "2021 global air passenger totals show improvement from 2020, but still only half pre-pandemic levels," [Online]. Available: https://www.icao.int/Newsroom/Pages/2021-global-air-passenger-totals-show-improvement.aspx. [Accessed 01 2023].

[3] Leeham News and Analysis, "Congestion costs billions, but airlines show little concern," [Online]. Available: https://leehamnews.com/2019/03/21/congestion-costs-billions-but-airlines-show-little-concern/. [Accessed 01 2023].

[4] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha and D. Abramson, "Scheduling Aircraft Landings—The Static Case," *Transportation Science*, 2000.

[5] J. E. Beasley, "OR-Library," June 1990; Last update: February 2018. [Online]. Available: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/airlandinfo.html. [Accessed 01 2023].

[6] M. Wallace, Principles and Practice of Constraint Programming - CP 2004, Springer, 2004.