SAAD: Aircraft Landing Problem - Static case with multi-runways

Ricardo Vilaça[1], Pedro Gouveia[1], and João Pedro Pêgo[1][0000−0002−7574−8879]

Faculdade de Engenharia da Universidade do Porto, Portugal
www.fe.up.pt
{up199502401, up202100813, up201308080 }@up.pt

Abstract. This paper aims to describe the achieved results on applying an optimization problem to a sequencing aircraft landing case study, that was designed, implemented and tested with two different search strategies: mathematical integer programming (MIP) model and a constraint programming (CP) model, to reach optimal solutions, using technologies mentioned in SAAD classes.

The chosen topic "Aircraft landing" consists in deciding on assigning a landing sequence to aircrafts, in one or more runways, meeting the mandatory predefined separation times for each type of aircraft. This approach observed a static sequence Aircraft landing problem.

Each landing sequence in a determined runway – or multiple runways –, lies within a predefined time slot, ensuring the most effective use of the available runways allowing the minimization of costs involved when deviation from the target landing time occurs.

To reach optimal solutions to the problem the DOCplex mp and cp solvers were used. All computational results are presented in this document.

Regarding multiple runaways sequence landing problem, using CP model, was developed an original approach that was proven very efficient considering the obtained feasible results.

Keywords: aircraft landing problem · scheduling · multiple runaways.

1 Introduction

The present document reports the work of the authors for the course unit Analytical decision support systems. The theme chosen relates to the scheduling of aircraft landings. The aircraft landing problem (ALP) has been thoroughly studied in the past decades. For this work we have used the data sets available in the OR-Library, which were used in the works [1,2] by J. E. Beasley.

The ALP is a scheduling problem where one must define the "best" sequence of landings for a given set of planes. Each plane has a target time of landing, an earliest time of landing and a latest time of landing. The objective is to chose the sequence of landings which allows for the least amount deviation to the target times. If a plane lands earlier or later than the target time, it incurs in a cost (due to excessive speed or because of waiting times). The landing of two consecutive planes must be such that a given separation time is respected. The separation time is necessary to guarantee that turbulence generated by one landing will not affect the next landing. The objective function is thus to minimize the costs derived from not landing planes on their target times, while respecting the separation times and landing windows.

Two cases were studied by Beasley et. al.: the static case and the dynamic case. In the static case, the landing targets are predefined. The dynamic case, is closer to reality, since the target times of the planes vary along the day. Due to time restrictions the authors decided to compute the static case, with multiple runaways, which corresponds to the work done in [1], where eight case studies where considered, involving from 10 to 50 planes.

A mixed-integer linear programming (MILP) and a constrain programming (CP) model were developed, using python DOCplex, and will be presented next.

2 MILP model

The MILP model was created following the work of [1]. The model starts by defining the relation between decision variables (e.g. if plane i lands before plane j the separation time between the two planes S_{ij} must be guaranteed).

$P =$ the number of planes
$E_i =$ the earliest landing time for plane i $(i = 1, \ldots, P)$
$L_i =$ the latest landing time for plane i $(i = 1, \ldots, P)$
$T_i =$ the target (preferred) landing time for plane i $(i = 1, \ldots, P)$
$S_{ij} =$ the separation time (≥ 0) between plane i landing and plane j landing (where plane i lands before plane j), $i, j = 1, \ldots, P; i \neq j$
$g_i =$ the penalty cost (≥ 0) per unit of time for landing before the target time $T_i =$ for plane i $(i = 1, \ldots, P)$
$h_i =$ the penalty cost (≥ 0) per unit of time for landing after the target time
$T_i =$ for plane i $(i = 1, \ldots, P)$
$R =$ the number of runaways

$$\delta_{ij} = \begin{cases} 1 \text{ if plane } i \text{ lands before plane } j \ (i, j = 1, \ldots, P; i \neq j) \\ 0 \text{ otherwise} \end{cases}$$

$$z_{ij} = \begin{cases} 1 \text{ if planes } i \text{ and } j \text{ land on the same runaway } (i, j = 1, \ldots, P; i \neq j) \\ 0 \text{ otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1 \text{ if plane } i \ (i = 1, \ldots, P) \text{ lands on runaway } r \ (r = 1, \ldots, R) \\ 0 \text{ otherwise} \end{cases}$$

Three decision variables were considered, which correspond to the landing time of planes and the deviation from the target time (before or after).

Decision variables:
$x_i =$ the landing time for plane i $(i = 1, \ldots, P)$
$\alpha_i =$ how soon plane i $(i = 1, \ldots, P)$ lands before T_i
$\beta_i =$ how soon plane i $(i = 1, \ldots, P)$ lands after T_i

The objective function is to minimize the costs associated to the deviations α_i and β_i. The penalty costs per unit of time must be provided.

Objective function:

$$\text{minimize} \sum_{i=1}^{P} (g_i \alpha_i + h_i \beta_i) \tag{1}$$

Finally, the solution for the landing times must be subject to several constrains that guarantee that both the landing window and the separation times will be met:

$$E_i \leq x_i \leq L_i \qquad i = 1, \ldots, P \tag{2}$$

$$\alpha_i \geq T_i - x_i \qquad i = 1, \ldots, P \tag{3}$$

$$0 \leq \alpha_i \leq T_i - E_i \qquad i = 1, \ldots, P \tag{4}$$

$$\beta_i \geq x_i - T_i \qquad i = 1, \ldots, P \tag{5}$$

$$0 \leq \beta_i \leq T_i - L_i \qquad i = 1, \ldots, P \tag{6}$$

$$x_i = T_i - \alpha_i + \beta_i \qquad i = 1, \ldots, P \tag{7}$$

$$\sum_{r=1}^{R} y_{ir} = 1 \qquad i = 1, \ldots, P \tag{8}$$

$$z_{ij} = z_{ji} \qquad i, j = 1, \ldots, P, j > i \tag{9}$$

$$z_{ij} \geq y_{ir} + y_{jr} - 1 \qquad \begin{array}{l} i, j = 1, \ldots, P; j > i \\ r = 1, \ldots, R \end{array} \tag{10}$$

$$x_j \geq x_i + S_{ij} z_{ij} - (L_i + S_{ij} - E_j)\delta_{ji} \qquad i, j = 1, \ldots, P \tag{11}$$

Constrains (2) to (7) are self-explanatory. condition (8) guarantees that a plane cannot land in more than one runway. Constrain (9) guarantees that if plane i lands in the same runway than plane j vice-versa is also valid. Constrain (10) guarantees that if there is any runaway r for which $y_{ir} = y_{jr} = 1$ then $z_{ij} = 1$. If $z_{ij} = 0$ then the planes i and j cannot land on the same runaway. Finally, constrain (11) guarantees that the Separation time between plane i and plane j must be respected.

3 CP model

3.1 Single runway (1R)

The constrain programming model is far simpler and more intuitive. Due to the complexity (many hours spent trying) of considering multiple runways and upon consultation with the teacher, the authors decided to just model the single runway case. The decision variables and the objective function remain the same. In terms of constrains, only four were used, with a slight change in the definition of α_i and β_i.

Decision variables:

x_i = the landing time for plane i $(i = 1, \ldots, P)$
α_i = how soon plane i $(i = 1, \ldots, P)$ lands before T_i
β_i = how soon plane i $(i = 1, \ldots, P)$ lands after T_i

Objective function:

$$\text{minimize} \sum_{i=1}^{P} (g_i \alpha_i + h_i \beta_i) \tag{12}$$

Constrains:

$$E_i \le x_i \le L_i \qquad , i = 1, \ldots, P \tag{13}$$
$$\alpha_i = \max(0, T_i - x_i) \qquad , i = 1, \ldots, P \tag{14}$$
$$\beta_i = \max(0, x_i - T_i) \qquad , i = 1, \ldots, P \tag{15}$$
$$x_j \ge x_i + S_{ij} \qquad , i = 1, \ldots, P \tag{16}$$

3.2 Multiple runways (mR)

P = the number of planes
R = the number of runways
$n_{slots} = \mod (P/R) + 1$ the number of slots in each runway
$fP = n_{slots} \cdot R - P$ the number of fictitious planes
$nP = P + fP$ the total number of planes
E_i = the earliest landing time for plane i $(i = 1, \ldots, nP)$
L_i = the latest landing time for plane i $(i = 1, \ldots, nP)$
T_i = the target (preferred) landing time for plane i $(i = 1, \ldots, nP)$
S_{ij} = separation time, $i, j = 1, \ldots, P; i \neq j$
g_i = the penalty cost for landing before the target time $(i = 1, \ldots, nP)$
h_i = the penalty cost for landing after the target time $(i = 1, \ldots, nP)$
T_i = for plane i $(i = 1, \ldots, nP)$
$E_i = min_E = \min(E_i | i = 1, \ldots, P)$ for $(i = P + 1, \ldots, nP)$
$L_i = max_L = \max(L_i | i = 1, \ldots, P)$ for $(i = P + 1, \ldots, nP)$
$T_i = \frac{max_L + min_E}{2}$ for $(i = P + 1, \ldots, nP)$
$g_i = h_i = 0$ for $(i = P + 1, \ldots, nP)$
$SP_s = S_{ij}$, where $s = i + nP \cdot j$ $(s = 1, \ldots, nP * nP)$

S_{ij} is the same as for the MILP for the real planes $(i, j = 1, \ldots, P)$ and 0 for the fictitious planes $(i, j = P + 1, \ldots, nP)$.

The basis for this model is to divide the runways in slots where planes will land. E.g., imagine one would like to allocate 10 planes to 3 runways. The number of slots in each runway will be modulo (10 / 3) + 1 = 4. In this case, one would have a total of 4 * 3 = 12 slots to allocate. Some of the slots will be occupied by the planes we want to allocate. The remaining are fictitious planes that cannot contribute to the cost function, so they have zero cost. The all_diff constrain will guarantee that we have each of the 12 planes (real and fictitious) allocated to the available slots. Suppose we have plane 3 in slot 5. The landing time of plane 3 will be $t_{land,slot_5}$, where $slot_5 = 3$.

Decision variables:
$slot_i =$ the slot where plane i lands $(i = 1, \ldots, nP)$
$t_{land,i} =$ the landing time for plane i $(i = 1, \ldots, nP)$
$\alpha_i =$ how soon plane i $(i = 1, \ldots, P)$ lands before T_i
$\beta_i =$ how soon plane i $(i = 1, \ldots, P)$ lands after T_i

Objective function:

$$\text{minimize} \sum_{i=1}^{P} (g_i \alpha_i + h_i \beta_i) \tag{17}$$

Constrains:

$$all\_diff(slot) \tag{18}$$

$$E_i \leq x_i \leq L_i \qquad , i = 1, \ldots, P \tag{19}$$

$$\alpha_i = \max(0, T_i - x_i) \qquad , i = 1, \ldots, P \tag{20}$$

$$\beta_i = \max(0, x_i - T_i) \qquad , i = 1, \ldots, P \tag{21}$$

$$t_{land,j} \geq t_{land,i} + S_{ij} \qquad , i = 1, \ldots, P \tag{22}$$

4 Results

The results of the models are presented in Table 1. The values of the objective function (cost) for the MILP model are identical to the ones of the original publication [1]. Since 22 years have past since the first publication, the solver times are much smaller in the present simulations. The results for the CP model (R=1) are identical to the MILP model.

The multiple runways model was developed very close to the deadline so we did not have time to test all cases. For Case 1, with 10 planes, the optimal solutions were obtained with the following solving times: $R = 1$ - 7.169 s; $R = 2$ - s; $R = 3$ - 0.121 s.

5 Source code

The source code is presented together with this report in the form of a jupyter notebook. To run the code DOCplex must be locally installed. The python notebook will fetch the data sets from the OR-Library and run the models for the selected cases. To run the first 8 cases, the code takes between 20-30 minutes.

Table 1. Model results

			MILP					CP 1R	
Case	R	P	n.c.var	n.b.var	n.const	value	sol. time [s]	value	sol. time [s]
1	1	10	30	210	320	700	0.109	700.0	0.259
1	2	10	30	220	365	90	0.093	–	–
1	3	10	30	230	410	0	0.015	–	–
2	1	15	45	465	705	1480	0.109	1480.0	3.384
2	2	15	45	480	810	210	0.093	–	–
2	3	15	45	495	915	0	0.031	–	–
3	1	20	60	820	1240	820	0.125	820.0	0.566
3	2	20	60	840	1430	60	0.110	–	–
3	3	20	60	860	1620	0	0.046	–	–
4	1	20	60	820	1240	2520	1.657	2520.0	300.051
4	2	20	60	840	1430	640	0.829	–	–
4	3	20	60	860	1620	130	0.250	–	–
4	4	20	60	880	1810	0	0.047	–	–
5	1	20	60	820	1240	3100	6.437	3100.0	300.045
5	2	20	60	840	1430	650	4.468	–	–
5	3	20	60	860	1620	170	1.234	–	–
5	4	20	60	880	1810	0	0.062	–	–
6	1	30	90	1830	2760	24442	0.016	–	–
6	2	30	90	1860	3195	554	0.296	–	–
6	3	30	90	1890	3630	0	0.047	–	–
7	1	44	132	3916	5896	1550	0.141	1550.0	47.820
7	2	44	132	3960	6842	0	0.031	–	–
8	1	50	150	5050	7600	1950	0.563	1950.0	300.191
8	2	50	150	5100	8825	135	1.609	–	–
8	3	50	150	5150	10050	0	0.750	–	–

R - n. runways; P - n. planes; n.c.var - n. continuous var.;
n.b.var - n. binary var.; n. const. - n. constrains;
sol. time - solver time; value - objective function value

6 Conclusions

In this work the aircraft landing problem (ALP) was solved using mixed integer linear programming and constrain programming. Three models were developed to solve the static ALP: multiple runways MILP; single runway CP and multiple runways CP. All models were able to reach the optimal solution computed by Beasley et.al. [1]. Overall, the MILP model was more capable and faster to solve the problem. We had a lot of headaches and fun!

References

1. Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D.: Scheduling aircraft landings—the static case. Transportation Science **34**(2), 180–197 (2000). https://doi.org/10.1287/trsc.34.2.180.12302
2. Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D.: Displacement problem and dynamically scheduling aircraft landings. Journal of the Operational Research Society **55**(1), 54–64 (2004). https://doi.org/10.1057/palgrave.jors.2601650