A heuristic algorithm for the Aircraft Landing Problem

Amir Salehipour [a]**, Mehdi Ahmadian** [b]

[a]*ARC DECRA Fellow, University of Technology Sydney, NSW, Australia*
[b]*Department of Industrial Engineering, Bu-Ali Sina University, Iran*
Email: amir.salehipour@gmail.com

Abstract: The Aircraft Landing Problem is the problem of allocating an airport's runways (processing units) to arriving aircraft (competing tasks) as well as scheduling the landing time of aircraft, with the objective of minimizing total deviations from the target landing times. This work proposes a new approach to solve the Aircraft Landing Problem that includes separating sequencing and scheduling aircraft. While obtaining the optimal sequence of landing is very difficult, particularly for large instances, an optimal schedule for a given sequence can be delivered in polynomial time. We apply the construction algorithm of Salehipour et al. (2013) to construct an initial sequence of landing, and then we improve this sequence by developing a relaxation heuristic, which relaxes and re-optimizes the sequence. Computational experiments over a set of standard instances demonstrate the performance of the proposed approach.

Keywords: Aircraft Landing Problem, sequencing, heuristic, sequence relaxation

1 INTRODUCTION

The Aircraft Landing Problem (ALP) is an important class of scheduling problems. Optimal solution of an ALP has applications in the areas of transportation, providing significant benefits to systems experiencing long delays. The objective of the ALP is to minimize total deviations from the target (preferred) landing times by penalizing early and late landings. The amount of deviation from the target landing times between the earliest and latest landing times is the basis for determining the penalty.

Given a set of aircraft $I = \{1, 2, \ldots, n\}$, the penalty of late landing of aircraft $i \in I$ is given by $\alpha_i = \max(0, x_i - T_i)$ when aircraft i lands later than T_i, that is, if $T_i \leq x_i \leq L_i$, where x_i is the *scheduled* landing time of aircraft i, T_i is the target or preferred landing time of aircraft i and L_i is the latest landing time of aircraft i. The penalty of early landing of aircraft i is given by $\beta_i = \max(0, T_i - x_i)$ when aircraft i lands earlier than T_i, that is, if $E_i \leq x_i \leq T_i$, where E_i is the earliest landing time of aircraft i. Note that α_i holds if the decision variable x_i lies within the range $[T_i, L_i]$, while β_i holds if it lies within the range $[E_i, T_i]$. Thus $E_i \leq x_i \leq L_i$ for each aircraft i, and $[E_i, L_i]$ is the feasible ranges for the landing time of aircraft i. The reader may realize that capturing two penalties results in a piecewise objective function. In addition, there is a safety *separation* time $s_{ij} \in \mathbb{R}^+$ between every two aircraft $i, j \in I^2, i \neq j$, if they land immediately one after another. The separation time can be different for different pairs of aircraft. Hence, a change in the landing sequence would change the landing time, and that of the objective function (penalties of early and late landings).

Exact solutions developed by Beasley, Ernst and Krishnamoorthy, and Ghoniem and Farhadi can only solve small sized instances of the ALP (Beasley et al., 2000; Ernst et al., 1999; Ghoniem and Farhadi, 2015). Therefore, heuristics and meta-heuristics are practical options for large sized instances. For example, Pinol and Beasley applied Scatter Search and Bionomic Algorithms to instances with up to 500 aircraft (Pinol and Beasley, 2006). Salehipour et al. (2013) developed an approach to combine Simulated Annealing and Variable Neighborhood Search algorithms into one hybrid algorithm, and applied this to the same instances. Those studies found optimal solution for small instances with up to 50 aircraft, and high quality solutions for larger instances.

2 SOLUTION FRAMEWORK

The solution framework proposed in this study decomposes the Aircraft Landing Problem (ALP) into two problems of sequencing aircraft landings, and scheduling those landings. It is not difficult to show that if only one runway is available, which is the case of the presented study, and a sequence for aircraft landings is given, obtaining an optimal schedule for the landings can be performed in polynomial time. Therefore, it only suffices to obtain a sequence (ideally optimal) for aircraft landings because we can obtain an optimal landing schedule for this sequence by utilizing available solvers, e.g. CPLEX, and Gurobi.

We propose a heuristic algorithm for obtaining this sequence. The algorithm includes generating an initial sequence, and improving it by applying a local search algorithm.

In order to obtain an initial sequence, we apply the construction heuristic of Salehipour et al. (2013). This algorithm obtains an initial sequence by sorting aircraft on a non-decreasing order of their target landing times. Therefore, the first aircraft in the sorted list lands first, the second aircraft in the sorted list lands second, and so on.

In order to improve a given sequence, our proposed local search includes allowing a small part of the sequence to be modified. More precisely, we allow $[k\%], k \in \mathbb{Z}^+$ of aircraft to be relaxed from their position in the sequence (we call this the *relaxation neighborhood*). Hence, the optimization process (re-sequencing the relaxed sequence) can move these aircraft both forward and backward in the sequence. For optimization process, we use the solver CPLEX. Note that because only a percentage, i.e. $[k\%]$, of aircraft is relaxed, CPLEX is able to optimally re-sequence the relaxed part. We perform the relaxation neighborhood until the stopping criterion is met. The heuristic algorithm for the ALP is summarized in Algorithm 1.

3 COMPUTATIONAL RESULTS

We tested Algorithm 1 on 13 standard benchmark instances of the Aircraft Landing Problem (ALP), which are available from OR Library at `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`. The instances are ranging from small (10 aircraft) to large (500 aircraft). In this study, we set $k = 5$; therefore, we relax $[5\%]$ of aircraft in each iteration of the algorithm. The reason for choosing $k = 5$ is because we

---

Algorithm 1: A heuristic algorithm for the Aircraft Landing Problem (ALP).

Input: An integer program (IP) for ALP; parameter $k \in \mathbb{Z}^+$.
Output: S, a sequence of aircraft landings.

$S \leftarrow \texttt{sort}(T_i)$ (obtaining an initial sequence by sorting aircraft on a non-decreasing order of target landing times T_i);
while *the stopping condition is not met* **do**
    $\texttt{relax}()$ (relaxing the position of $[k\%]$ aircraft in S);
    $S' \leftarrow \texttt{optimize}()$ (optimizing the relaxed sequence by using the solver CPLEX);
    $\Delta \leftarrow \texttt{objective}(S') - \texttt{objective}(S)$;
    if $\Delta < 0$ **then**
        | $S \leftarrow S'$;
    end
end
Report S;

---

realized that this value of k results in very good quality solutions in a short time. In particular, we observed that larger values ($k > 5$) increase the computation time without significant impact on the solution quality, and smaller values ($k < 5$) result in increased number of optimization iterations with considerable amount of computation time. Furthermore, due to the penalty for both earliness and tardiness we observed that aircraft may not be re-sequenced too far from the current position both forward and backward. Also, we consider one runway. Generally speaking, the ALP with single runway is more difficult to solve than with multiple runways. The algorithm was coded in the programming language C++, and we implemented the solver CPLEX 12.5.0 (ILOG, 2012).

The outcomes of the proposed heuristic (Algorithm 1) have been illustrated in Table 1 and Table 2. In Table 1 we compared Algorithm 1 with the solver CPLEX and MRASP (multiple-runway aircraft sequencing problem) algorithm of Ghoniem and Farhadi (2015), and in Table 2 we compared it with the best heuristics available in the literature.

Let us first discuss the results presented in Table 1. The three methods obtain optimal solution for all instances with up to 50 aircraft. For the remaining five instances, while CPLEX average computation time is around 12 hours, and that of MRASP is 1 hour, the average computation time of Algorithm 1 is around 5 minutes (notice that CPLEX was run on a high performance machine with 10 threads of Intel®Xeon CPU E5-1650 at 3.50GHz, and 32 GB of memory, and it was not interrupted until it stopped due to lack of memory, whereas Algorithm 1 was run on a personal computer with only 1 thread of Intel®Core i5-4570S CPU at 2.90GHz, and 8 GB of memory). Within the computation times reported in Table 1, Algorithm 1 obtains the best known solution for 10 instances (these are highlighted), while the MRASP obtains only 8, and that within about 25 minutes. It is wroth mentioning that the average gap of Algorithm 1 is less than 1%.

In comparing Algorithm 1 with the best available heuristics in the literature, we chose two algorithms of Simulated Annealing + Variable Neighborhood Descent (SA+VND) (Salehipour et al., 2013), and Scatter Search (SS) (Pinol and Beasley, 2006). Here, one may realize that the maximum computation time of Algorithm 1 is much shorter than those algorithms, while it has a smaller average gap. In addition to these, Algorithm 1 obtains new best solutions for several instances, compared to those two algorithms.

4 CONCLUSION

Optimal solutions of the Aircraft Landing Problem (ALP) has applications in the areas of transportation, providing significant benefits to systems experiencing long delays. The objective of the ALP is to minimize total deviations, both earliness and tardiness, from the target (preferred) landing times. Following the difficulty of the problem, we proposed a new approach to solve the ALP by separating sequencing and scheduling decisions. The proposed heuristic algorithm generates a landing sequence for the ALP. Then, the optimal landing schedule for this sequence can be obtained by utilizing available solvers. We showed that while this approach is quick enough to be implemented in practice, it is very competitive to the available solution methods, in terms of obtaining good quality solutions.

Table 1. Comparing the outcomes of Algorithm 1 with the outcomes of solver CPLEX and MRASP (Ghoniem and Farhadi, 2015). The CPLEX was run on a high performance machine with 10 threads of Intel®Xeon CPU E5-1650 at 3.50GHz, and 32 GB of memory, and Algorithm 1 was run on a personal computer with only 1 thread of Intel®Core i5-4570S CPU at 2.90GHz, and 8 GB of memory.

Instance	n	z^*	CPLEX				MRASP (Ghoniem and Farhadi, 2015)			Algorithm 1		
			$z*$	Time(s)	Nodes left	Gap(%)	$z*$	Time(s)	Gap(%)	$z*$	Time(s)	Gap(%)
Airland1	10	700.00	700.00	0.66	0	0.00	700.00	0.28	0.00	**700.00**	0.22	0.00
Airland2	15	1480.00	1480.00	0.49	0	0.00	1480.00	0.12	0.00	**1480.00**	1.71	0.00
Airland3	20	820.00	820.00	0.39	0	0.00	820.00	0.12	0.00	**820.00**	2.69	0.00
Airland4	20	2520.00	2520.00	5.12	0	0.00	2520.00	1.45	0.00	**2520.00**	2.78	0.00
Airland5	20	3100.00	3100.00	20.44	0	0.00	3100.00	4.14	0.00	**3100.00**	3.07	0.00
Airland6	30	24442.00	24442.00	0.10	0	0.00	24442.00	0.08	0.00	**24442.00**	0.21	0.00
Airland7	44	1550.00	1550.00	0.86	0	0.00	1550.00	0.14	0.00	**1550.00**	0.61	0.00
Airland8	50	1950.00	1950.00	0.98	0	0.00	1950.00	0.50	0.00	**1950.00**	22.58	0.00
Airland9	100	5611.70	5611.70	35439.15	114,888,161	15.08	5611.99	3600.00	0.01	**5611.70**	49.84	0.00
Airland10	150	12292.20	12292.20	58182.04	98,406,232	50.77	12310.70	3600.00	0.15	12359.00	298.25	0.54
Airland11	200	12418.30	12418.32	30713.95	77,355,422	36.80	12418.32	3600.00	0.00	**12418.30**	286.69	0.00
Airland12	250	16152.73	16152.73	33798.37	61,092,575	44.07	16302.28	3600.00	0.93	16618.20	282.65	2.88
Airland13	500	37268.12	37268.12	51666.16	38,294,894	53.22	37294.76	3600.00	0.07	39620.80	305.45	6.31
Min				0.10		0.00		0.08	0.00		0.21	0.00
Mean				16140.67		15.38		1385.14	0.09		96.67	0.75
Max				58182.04		53.22		3600.00	0.93		305.45	6.31

Table 2. Comparing the outcomes of Algorithm 1 with the best heuristics available in the literature.

Instance	n	z^*	SA+VND (Salehipour et al., 2013)			SS (Pinol and Beasley, 2006)			Algorithm 1		
			$z*$	Time(s)	Gap(%)	$z*$	Time(s)	Gap(%)	$z*$	Time(s)	Gap(%)
Airland1	10	700.00	700.00	0.00	0.00	700.00	4.00	0.00	**700.00**	0.22	0.00
Airland2	15	1480.00	1480.00	1.59	0.00	1480.00	6.00	0.00	**1480.00**	1.71	0.00
Airland3	20	820.00	820.00	1.78	0.00	820.00	8.00	0.00	**820.00**	2.69	0.00
Airland4	20	2520.00	2520.00	1.98	0.00	2520.00	8.00	0.00	**2520.00**	2.78	0.00
Airland5	20	3100.00	3100.00	1.85	0.00	3100.00	9.00	0.00	**3100.00**	3.07	0.00
Airland6	30	24442.00	24442.00	2.12	0.00	24442.00	158.00	0.00	**24442.00**	0.21	0.00
Airland7	44	1550.00	1550.00	2.68	0.00	1550.00	195.00	0.00	**1550.00**	0.61	0.00
Airland8	50	1950.00	1950.00	7.10	0.00	1950.00	42.00	0.00	**1950.00**	22.58	0.00
Airland9	100	5611.70	6091.88	11.59	8.56	6091.88	119.00	8.56	**5611.70**	49.84	0.00
Airland10	150	12292.20	12329.31	20.12	0.30	12329.31	227.00	0.30	12359.00	298.25	0.54
Airland11	200	12418.30	12418.32	24.17	0.00	12418.32	256.00	0.00	**12418.30**	286.69	0.00
Airland12	250	16152.73	16209.78	219.03	0.35	16209.78	381.00	0.35	16618.20	282.65	2.88
Airland13	500	37268.12	39867.33	566.82	6.97	44832.38	1237.00	20.30	39620.80	305.45	6.31
Min				0.00	0.00		4.00	0.00		0.21	0.00
Mean				66.22	1.25		203.85	2.27		96.67	0.75
Max				566.82	8.56		1237.00	20.30		305.45	6.31

REFERENCES

Beasley, J. E., M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson (2000). Scheduling aircraft landingsthe static case. *Transportation science 34*(2), 180–197.

Ernst, A. T., M. Krishnamoorthy, and R. H. Storer (1999). Heuristic and exact algorithms for scheduling aircraft landings. *Networks 34*(3), 229–241.

Ghoniem, A. and F. Farhadi (2015). A column generation approach for aircraft sequencing problems: a computational study. *Journal of the Operational Research Society 66*(10), 1717–1729.

ILOG, I. (2012, 01). *IBM ILOG CPLEX V12.4.0: User's manual for CPLEX.*

Pinol, H. and J. Beasley (2006). Scatter search and bionomic algorithms for the aircraft landing problem. *European Journal of Operational Research 171*(2), 439 – 462.

Salehipour, A., M. Modarres, and L. M. Naeni (2013). An efficient hybrid meta-heuristic for aircraft landing problem. *Computers & Operations Research 40*(1), 207 – 213.