Iterated Local Search Algorithm for single and double runway aircraft scheduling problems in Terminal Area

Hongyun Xu, Quan OuYang
School of Mathematics & Computer Science, JiangHan University, Wuhan 430056, China
oyq8888@163.com (Corresponding author: Quan OuYang)

Abstract. We present an Iterated Local Search (ILS) algorithm for solving the single and double runway aircraft scheduling problems in terminal area. The main ideas and implementation methods of each step are described in detail. Applying the ILS algorithm to solve the instance used in the literature and compare the results of ILS with those of previous 4 algorithms. For the single runway problem, the ILS can find the previous best known result. For the double runway problem, the ILS can find a result better than the previous best known result.

1. Introduction
The aircraft scheduling problem in terminal area can be formally stated as follows: There are $n$ aircraft entering the terminal area waiting for landing, and there are three types of aircraft: small (S), large (L), and heavy (H). Each aircraft $a$ has an estimated time of arrival $ETA(a)$. There is a minimum separation $C(a, b)$ for any two aircraft $a$ and $b$ that will be incurred if aircraft $b$ landed immediately after aircraft $a$. Let $P = \{P_1, P_2, ..., P_n\} (P_i \neq P_j, i \neq j)$ represent a landing sequence of the $n$ aircraft, where $P_i$ denotes the aircraft landed in the $i$th position of $P$. Then the scheduled time of arrival of $P_i$ is $STA(P_i)$ and which is calculated as:

$$STA(P_i) = \max \{ETA(P_i), STA(P_{i-1}) + C(P_{i-1}, P_i)\}$$

The delay of $P_i$ is $d(P_i)$ and which is calculated as:

$$d(P_i) = STA(P_i) - ETA(P_i)$$

The aircraft scheduling problem in terminal area is to find an aircraft sequence that the total delay of which is minimal. For the single runway problem, the objective function $F(P)$ is:

$$F(P) = \min \left\{ \sum_{i=1}^{n} [STA(P_i) - ETA(P_i)] \right\}$$

For the double runway problem, the objective function $F'(P)$ is:

$$F'(P) = \min \left\{ \sum_{i=1}^{n} [STA(P_i) - \min\{ETA_1(P_i), ETA_2(P_i)\}] \right\}$$

In which, $ETA_1(P_i)$ and $ETA_2(P_i)$ represent the estimated time of arrival of $P_i$ on the runway #1 and runway #2 of the double runway, respectively.

Aircraft scheduling belongs to the category of scheduling problem and it’s an NP-hard problem when the scale of the problem reaches a certain level. When solving the aircraft scheduling problem,
the most common method is the first-come-first-served (FCFS) algorithm, which relies on the order of the estimated time of arrival (ETA) to determine the landing sequence of the aircraft. It might give rise to a large delay because it is without any optimization method. Due to the obvious defects of the FCFS algorithm, scholars have done a lot of research on aircraft scheduling optimization algorithms in recent years and achieved some results. In [1], Erzberger H et al. used a Time Advance (TA) algorithm to handle the aircraft scheduling problem. Beasley J et al. [2] adopted the Constrained Position Shift (CPS) algorithm. Minghua H [3] used the dynamic sorting algorithm. Xiaohao X et al.[4] used the fuzzy comprehensive evaluation method to solve the aircraft scheduling problem. In [5, 6], both Cheng VHL and Hansen JV used the genetic algorithms to solve the aircraft scheduling problem. However, as the problem scales gradually expanding and the objective function continuously complicated, the CPS algorithm has poor real-time performance, which is difficult to meet the practical application. The dynamic sorting algorithm needs to constantly adjust the aircraft’s order and the controller’s workload is increased correspondingly. The function of fuzzy comprehensive evaluation method is greatly affected by the practical problem, which is difficult to determine [7]. The advantages and disadvantages of genetic algorithm designing usually depend on chromosome coding, initial population, design of selection strategy, selection of crossover operators, mutation, and design of fitness function and so on. However, only the best combination of these series steps could get better optimization results.

In this paper, we present an iterated local search (ILS) algorithm for solving the single and double runway aircraft scheduling problems in terminal area. We carry out experiment on public benchmark instances of the aircraft scheduling problems in terminal area in the literature and compare our computational results with the best known results obtained by previous 4 algorithms. Our ILS algorithm’s results are equal to or better than the previous algorithm’s results.

2. Iterated Local Search Algorithm for Solving aircraft scheduling in Terminal Area

Iterated local search (ILS) is a simple but widely used heuristic algorithm that uses the local search technique and the perturbation mechanism iteratively to search the solution space during the work process. The iterated local search algorithm usually consists of 5 parts: (1) Generate Initial Solution; (2) Local Search; (3) Evaluation of solution; (4) Perturbation; 5. Acceptance Criterion[8,9]. The basic framework of the iterated local search algorithm is shown in Algorithm 1.

Algorithm 1. The pseudocode of the ILS algorithm

Step 1: P ← Generate initial solution
Step 2: P' ← local search (P)
Step 3: repeat
Step 4: P* ← perturbation (P')
Step 5: P*' ← local search (P*)
Step 6: P' ← acceptance criterion (P', P*)
Step 7: until stop condition met

2.1. Generation of initial solution

For the proposed ILS algorithm, the generations of initial solution are: For the single runway problem, generating the initial aircraft sequence randomly. The aircraft sequence only contains the flight number, which is expressed as (HC0 LC1 HC2 HC3 SC4 HC5 LC6 HC7 HC8 SC9), where HC0, LC1, ..., SC9 are flight numbers and the first letter of them indicates the type of aircraft; For the double runway problem, generating a set of aircraft sequences randomly which is consist of the flight numbers and the runway number and denoted as

\[
\begin{bmatrix}
HC0 & HC1 & HC2 & HC3 & SC4 & HC5 & LC6 & HC7 & HC8 & SC9 \\
1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 2 & 1
\end{bmatrix}
\]

and then selecting an aircraft sequence with the minimal delay from these sequences as the initial solution.
2.2. Neighborhood structure and its evaluation strategy

For the local search algorithm, the core mission of which is to construct a good neighborhood structure. The quality of the neighborhood structure is the main factor affecting the efficiency of the local search algorithm. The aircraft scheduling problem in terminal area studied in this paper has very similar characteristics with the single machine scheduling problem. In particularly, the single runway aircraft scheduling problem in terminal area and the single machine scheduling problem almost have the same working scene. Therefore, the Block Move neighborhood structure proposed in [10] is adopted in the ILS algorithm, which is defined as follows: In the current aircraft sequence \( P = \{P_1, P_2, ..., P_n\} \), randomly select \( l \) consecutive elements starting from position \( i \) and the target position \( j \), then move the Block of length \( l \) to the position \( j \) of the current sequence.

This paper mainly studies the single and double runway aircraft scheduling in terminal area. In view of the fact that these two problems have different characteristics when using the Block Move neighborhood structure, different solutions are needed to evaluate the solution. For the single runway problem, it has the same work situation as the single machine scheduling problem, so the Block based fast evaluation strategy, incremental evaluation, in the literature [10] is directly used in the ILS algorithm. For the double runway problem, since there are two runways, it is difficult to use the Block based incremental evaluation strategy in [10]. Therefore, the solution is evaluated by calculating its objective function value directly. Since the number of flights taking off and landing per unit time is predictable, and the landing flight is assigned to two different runways, so it is feasible to evaluate the solution by calculating its objective function value directly.

2.3. Perturbation operator

The perturbation operator is the main component of the ILS algorithm architecture. It must solve two problems: How to perturb and how strong is the perturbation? The processes of the perturbation in the ILS algorithm are as follows: (1) Determine the perturbation object. Generating a random number which is divided by 2, when the remainder is 0 the global optimal solution is selected for the perturbation object, otherwise, the local optimal solution is selected; (2) Determine the perturbation strength. In this paper, the perturbation strength is \( 0.3 \times N \times \text{rand}() \times (0.2 \times N) \) where \( N \) is the number of aircraft. It is characterized by using the random function to limit the perturbation strength in a certain range. Thus, the situations like, the purpose of perturbation is not achieved because the number of perturbation is too few and the waste of the computation time because the number of perturbation is too many, will not happen; (3) Determine the perturbation operation. The ILS algorithm selects the Block Move movement as perturbation operation.

2.4. Acceptance criterion and stop condition

The acceptance criterion is used to determine whether the new solution can continue to iterate as the current solution. Acceptance criterion can be well used to balance the diversity and concentration of search. After completing a local search process, the ILS algorithm compares the new local optimal solution \( P' \) with the current best solution \( P \). Judging the merits and demerits of solutions are directly compared the objective function value of them. The aircraft scheduling in terminal area is a minimization problem, the solution with a smaller objective function value is better. If \( f(P') \) is less than the current best solution \( f(P) \), then the current best solution \( P \) is replaced by \( P' \), i.e. if \( f(P') < f(P) \), then \( P = P' \). The stopping condition of the ILS algorithm is when the time limit or the number of iterations limit is reached the ILS algorithm stops.

3. Computational results

This paper uses 10 aircraft data provided in the literature [11] for evaluating the performance of the proposed ILS algorithm. Flight number and types are as follows:

\[
\text{FLIGHT\_NUM} = [\text{HC0}, \text{LC1}, \text{HC2}, \text{HC3}, \text{SC4}, \text{HC5}, \text{LC6}, \text{HC7}, \text{HC8}, \text{SC9}]
\]

\[
\text{AIRCRAFT\_TYPE} = [\text{H}, \text{L}, \text{H}, \text{S}, \text{H}, \text{L}, \text{H}, \text{S}]
\]
According to ICAO regulations, there are three types of aircraft, heavy (H), large (L), and small (S). The minimum wake separation standard $C$ between different types of aircraft is represented by a $3 \times 3$ time matrix (in min):

\[
\text{SEPARATION CONSTRAINTS} = \begin{bmatrix}
1.0 & 1.5 & 2.0 \\
1.0 & 1.5 & 1.5 \\
1.0 & 1.0 & 1.0
\end{bmatrix}
\]

The rows represent the type of aircraft in front and the columns represent the type of aircraft followed. Both the row and column represent the type of aircraft in order of H, L, and S.

Our ILS algorithm was programmed in C++ and all the experiments were carried out on a computer with a core i5 1.4 GHz CPU and 4.0 GB RAM. We solved the single runway and double runway instances independently for 10 times using different random seeds, subject to a time limit of 1 CPU second.

3.1. Single runway
For the single runway, the estimated time of arrival is as follows:

\[
\text{ETA} = \begin{bmatrix}
11.0 & 15.0 & 6.0 & 6.0 & 9.0 & 7.0 & 15.0 & 6.0 & 6.0 & 9.0
\end{bmatrix}
\]

The results of the proposed ILS algorithm are compared with those of FCFS and MA [11] algorithms by using the above data, which are shown in Table 1.

| No. | FCFS ETA | STA | MA ETA | STA | ILS ETA | STA |
|-----|----------|-----|--------|-----|---------|-----|
| HC2 | 6.0      | 6.0 | 6.0    | 6.0 | 6.0     | 10.0|
| HC3 | 6.0      | 7.0 | 6.0    | 7.0 | 6.0     | 6.0 |
| HC7 | 6.0      | 8.0 | 6.0    | 8.0 | 6.0     | 7.0 |
| HC8 | 6.0      | 9.0 | 6.0    | 9.0 | 6.0     | 8.0 |
| HC9 | 7.0      | 10.0| 7.0    | 10.0| 7.0     | 9.0 |
| SC4 | 9.0      | 12.0| 9.0    | 13.0| 9.0     | 13.0|
| SC9 | 9.0      | 13.0| 9.0    | 14.0| 9.0     | 14.0|
| HC0 | 11.0     | 14.0| 11.0   | 11.0| 11.0    | 11.0|
| LC1 | 15.0     | 15.5| 15.0   | 15.0| 15.0    | 15.0|
| LC6 | 15.0     | 17.0| 15.0   | 16.5| 15.0    | 16.5|

Delay 21.5 mins 19.5 mins 19.5 mins

As can be seen from Table 1, the total delay of FCFS, MA, and ILS are 21.5, 19.5, and 19.5 minutes, respectively. The total delay of ILS is the same as the MA’s and 3 minutes less than the FCFS’s.

3.2. Double runway
For the double runway, the estimated time of arrival is as follows:

\[
\text{ETA} = \begin{bmatrix}
11.0 & 15.0 & 6.0 & 6.0 & 9.0 & 7.0 & 15.0 & 6.0 & 6.0 & 9.0 \\
10.0 & 17.0 & 7.0 & 12.0 & 6.0 & 17.0 & 7.0 & 7.0 & 12.0
\end{bmatrix}
\]

\[
\text{ETA}(i) = \min\{\text{ETA}[0][i], \text{ETA}[1][i]\}
\]

Using the above data, the results of the proposed ILS algorithm are compared with those of FCFS and MA algorithms, which are shown in Table 2.
From Table 2, one clearly observes that the total delay of FCFS, MA, and ILS are 12.5, 6.5, and 5.0 minutes, respectively. The total delay of ILS is 5.0 mins, which are 7.5 mins and 1.5 mins less than that of FCFS and MA, respectively.

Table 2. The detailed results of ILS algorithm for double runway

| Runway #1 | Runway #2 | Runway #1 | Runway #2 | Runway #1 | Runway #2 |
|-----------|-----------|-----------|-----------|-----------|-----------|
| No. ETA STA | No. ETA STA | Code ETA STA | No. ETA STA | Code ETA STA |
| HC2 6.0 6.0 | HC5 6.0 6.0 | HC7 6.0 6.0 | HC2 6.0 6.0 | HC7 6.0 6.0 |
| HC3 6.0 7.0 | HC0 10.0 10.0 | HC5 6.0 7.0 | HC3 6.0 7.0 | HC0 10.0 10.0 |
| HC7 6.0 8.0 | SC4 9.0 9.0 | SC9 9.0 10.0 | HC8 6.0 8.0 | SC9 9.0 10.0 |
| HC8 6.0 9.0 | SC9 9.0 10.0 | HC0 10.0 10.0 | HC8 6.0 9.0 | SC9 9.0 10.0 |
| SC4 9.0 11.0 | LC1 15.0 15.0 | LC6 15.0 16.5 | SC9 9.0 12.0 | LC1 15.0 15.0 |
| SC9 9.0 12.0 | LC6 15.0 16.5 | LC1 15.0 15.0 | SC9 9.0 12.0 | LC6 15.0 16.5 |
| LC1 15.0 15.0 | | | | |
| LC6 15.0 16.5 | | | | |

Delay: 12.5 mins Delay: 6.5 mins Delay: 5.0 mins

3.3. Comparison with other reference algorithms

We compare our results under the time limit of 1 second with the several previous algorithms in the literature. We compared the best, worse, and average results of ILS with those of FCFS, ACO, SA, and MA in table 3.

Table 3. Comparisons of ILS with other 4 reference algorithms

| Single runway | Double runway |
|---------------|---------------|
| best | worst | average | best | worst | average |
| FCFS 21.5 | 21.5 | 21.5 | 12.5 | 12.5 | 12.5 |
| ACO 19.5 | 21.5 | 19.5 | 6.5 | 9.0 | 7.0 |
| SA 19.5 | 23.5 | 21.5 | 7.0 | 23.5 | 10.5 |
| MA 19.5 | 19.5 | 19.5 | 6.5 | 7.0 | 6.5 |
| ILS 19.5 | 19.5 | 19.5 | 5.0 | 10.5 | 6.75 |

It can be seen from Table 3 that the ILS algorithm proposed in this paper can find the previous best known result 19.5 mins for the single runway problem. For the double runway problem, the best result of ILS algorithm is 5 mins, which improves the previous best known result and is 1.5 mins less than the previous best known result 6.5 mins.

4. Conclusion

This paper studies the application of ILS algorithm in solving single runway and double runway aircraft scheduling problems in terminal area. We test the proposed ILS algorithm and compare the results with the other 4 reference algorithms in the literature. The results show that the ILS algorithm can find a better solution when the program scale is small. The ILS algorithm is characterized by its easy implementation and good real-time performance, which can meet the practical application and reduce the workload of controllers.

Acknowledgments

This work was supported by the Doctoral Scientific Research Foundation of JiangHan University, WuHan (Grant no. 1008-06280004).

References

[1] Erzberger H, Nedell W 1989 Design of automated system for management of arrival traffic
[2] Beasley J, Krishnamoorthy M, Abramson D 2000 Scheduling aircraft landings: the static case. J. Transportation Science. 34(2) 180-197.

[3] Minghua H 2000 Dynamic Queue Algorithm in Passive Air Traffic Flow Management. J. Journal of Nanjing University of Aeronautics & Astronautics. 32(1) 85-90.

[4] Hao, X. X., & Jun, H. B 2001 Study of fuzzy integrated judge method applied to the aircraft sequencing in the terminal area. J. Acta Aeronautica Et Astronautica Sinica. 22(3) 259-261.

[5] Cheng V H L, Crawford L S, Menon P K 2002 Air Traffic Control Using Genetic Search Techniques. C. IEEE International Conference on Control Applications. IEEE.

[6] Hansen J V 2004 Genetic search methods in air traffic control. J. Computers & Operations Research. 31(3) 445-459.

[7] Chang Huixian, QU Shiru 2010 Genetic Algorithm of Aircraft Sequencing in Terminal Area. J. Measurement & Control Technology. 29(7) 91-93.

[8] Dong X, Huang H, Chen P 2009 An iterated local search algorithm for the permutation flowshop problem with total flowtime criterion. J. Computers and Operations Research. 36(5) 1664-1669.

[9] Lourenço H R, Martin O C, Stützle T 2003 Iterated local search. Springer.

[10] Xu H, Li Zhipeng, Cheng T C E 2014imes to minimize total weighted tardiness. J. Journal of Scheduling. 17(3) 271-287.

[11] Wang Fei, Xu Xiao-hao, Zhang Jing 2008 Mixed artificial fish school algorithm of aircraft sequencing in terminal area. J. Journal of Transportation Engineering. 8(3).