Advances in Aircraft and Spacecraft Science, *Vol. 9, No. 4 (2022) 335-347*
https://doi.org/10.12989/aas.2022.9.4.335

Effective simulation-based optimization algorithm for the aircraft runway scheduling problem

Ali Wided* and Bouakkaz Fatima[a]

Department of Mathematics and Computer Science, Larbi Tebessi University, Tebessa, Algeria

(Received September 15, 2021, Revised September 15, 2022, Accepted September 30, 2022)

Abstract. Airport operations are well-known as a bottleneck in the air traffic system, putting growing pressure on the world's busiest airports to schedule arrivals and departures as efficiently as possible. Effective planning and control are essential for increasing airport efficiency and reducing aircraft delays. Many algorithms for controlling the arrival/departure queuing area are handled, considering it as first in first out queues, where any available aircraft can take off regardless of its relative sequence with other aircraft. In the suggested system, this problem was compared to the problem of scheduling n tasks (plane takeoffs and landings) on a multiple machine (runways). The proposed technique decreases delays (via efficient runway allocation or allowing aircraft to be expedited to reach a scheduled time) to enhance runway capacity and decrease delays. The aircraft scheduling problem entails arranging aircraft on available runways and scheduling their landings and departures while considering any operational constraints. The topic of this work is the scheduling of aircraft landings and takeoffs on multiple runways. Each aircraft's takeoff and landing schedules have time windows, as well as minimum separation intervals between landings and takeoffs. We present and evaluate a variety of comprehensive concepts and solutions for scheduling aircraft arrival and departure times, intending to reduce delays relative to scheduled times. When compared to First Come First Serve scheduling algorithm, the suggested strategy is usually successful in reducing the average waiting time and average tardiness while optimizing runway use.

Keywords: aircraft scheduling; aircraft sequencing; minimizing delays; optimizing runway utilization; tardiness; waiting time

1. Introduction

The runway is interlinked between the terminal airspace and the airport network, and its capacity is typically considered the bottleneck of the turn-around process. The number of runways at most international hub airports' runway systems typically ranges from two to five. It's worth noting that new runway development may not be possible over the next five to ten years, as runway expansion is generally constrained by geographical restrictions and the limited land area available on the airport's surface. As a result, increasing runway engineering capacity is difficult. Apart from expanding runway physical capacity, we might pursue a systemic strategy to increase runway capacity.

As a result, many researchers and engineers are turning to a potential approach: better

*Corresponding author, Ph.D., E-mail: wided.ali@univ-tebessa.dz
[a]Ph.D. Student, E-mail: f_bouakkez@esi.dz

http://www.techno-press.org/?journal=aas&subpage=7 ISSN: 2287-528X (Print), 2287-5271 (Online)

scheduling the aircraft arrival sequence so that the runway can land as many aircraft as feasible in a given time. In this work, the optimization process is defined as aircraft sequencing and scheduling.

This paper is organized as follows. In Section 2, the problem description is presented. The aircraft sequencing and scheduling algorithms are proposed in Section 3. In Section 4, a numerical study is conducted to compare the performance of this new algorithm with the FCFS algorithm, while some conclusions are summarized in Section 5.

1.1 Aircraft sequencing and scheduling problem

In general, a real problem can only be represented as a version pure of a traditional operational research problem. However, the job scheduling problem is quite similar to an aircraft scheduling problem.

First analogy: job scheduling problem

Classically, a link has been established by Beasley *et al.* (2000) between the aircraft scheduling problems and the problems of job scheduling with sequence-dependent setup times. The following is the outline of the correspondence proposed by Fischetti and Salvagnin (2010):

- Runways are machines.
- Aircraft are Jobs.
- The minimum separation between two successive aircraft, *i* followed by *j* on the same runway, can be interpreted as the sum of the duration of jobi (occupation runway) and the setting time (idle time) between *i* and *j*.

Note that according to this analogy, the duration of the jobs is considered constant and equal (equal processing times) while the adjustment times are dependent on the sequence (sequence-dependent setup time).

The problem can be made more complex by adding time windows specific to the aircraft while keeping the analogy with the job scheduling problems.

- The earliest landing times of aircraft are the arrival or availability times of the jobs (release dates).
- The latest landing times correspond to the deadlines (due dates or deadlines).

1.2 Related works

The First-Come, First-Served (FCFS) scheduling algorithm is the most common approach used by controllers to sequence aircraft. The aircraft sequencing in this algorithm is determined using the First-Come-First-Served (FCFS) order. When the FCFS order is used, the aircraft that arrives at the Terminal Maneuvering Area(TMA) first is given landing priority. As a result, schedules at the start and end locations are only calculated to ensure that the minimum separation objectives are satisfied (Chandran *et al*. 2007).

The FCFS order has many advantages: 1) it is simple to implement, 2) it decreases air traffic controller workload, and 3) it is a reasonably fair sequencing approach, therefore it has been chosen as the baseline operation in many studies (Rathinam *et al*. 2008, Balakrishnan *et al*. 2006). However, it is rarely the best choice in terms of efficiency, and runway throughput, especially in congested airports. Also, the FCFS order may result in increased aircraft separation.

Because the TMA is such a complicated environment, aircraft resequencing from the FCFS order may increase the workload of air traffic controllers (Venkatakrishnan *et al.*1993).

Constrained Position Shifting (CPS) constraints are also incorporated in the second method to account for the limited flexibility in this quinquennial deviation from the FCFS order. An aircraft can modify its sequence with other aircraft under the CPS constraint by a specified maximum number of positions from the FCFS order. The maximum number of position shifts is specified by *k* between 1 and 3, resulting in a *k* CPS constraint (Balakrishnan *et al*. 2010).

The study (Pohl *et al*. 2021) proposed an optimization model for the problem of runway scheduling in the context of winter operations. The static and deterministic character of this method is a disadvantage. Because multiple aspects of the problem, such as aircraft time windows, aircraft parameters, and weather conditions, are subject to uncertainty and change; it may be beneficial to evaluate the stochasticity of the scenario.

This study (Bo *et al*. 2021) presented an optimization model for assigning a set of arrival and departure aircraft to multiple runways and finding their actual times while taking incursions into account. The time of fuzzy incursion with the use of artificial experience is employed to represent the uncertainty. This model also takes into account air traffic controllers' multiple-goal priority concerns. The study's main contributions are as follows: As the number of runways increases, the arrival flight delay time in this model improves over the traditional model, but it also generates significant delays for incoming and departing flights. Furthermore, when an incursion event happens, even if the arrival flight's scheduling priority is higher than the departure flight's, some arrival flights must choose to make an emergency landing at a nearby airport since their queuing time exceeds the flight time allowed by the remaining fuel. As a result, it's worth looking into the process of extending this model to choose diverted flights simultaneously using a point fusion program.

The proposed approach takes into consideration the drawbacks of the previous methods. We utilized a dynamic scheduler of aircraft to address the uncertainty and change in the airport environment. We tried to take into account several performance metrics for evaluating the proposed approach, such as average waiting time, tardiness, and average runway usage rate. Hard restrictions are considered in the article, such as the minimum separation time between aircraft, the prohibition of aircraft assignments before their scheduled time, and the prohibition of two specific aircraft being assigned at the same time.

2. Problem description

There are N jobs and M machines in the basic parallel machine scheduling problem. Each job must be completed on one of the machines within a fixed processing time. The challenge of parallel machine scheduling involves both resource allocation and sequencing. It assigns jobs to machines and sets the order in which those jobs are completed on the allocated machine. Machines can be identical, in which case each job's processing time is independent of the assigned machine; where each machine has a different speed with a known speed factor; or unrelated. Also, each job's processing time is dependent on the assigned machine without any particular relationship. The goal is to create a schedule that maximizes one or more performance metrics, such as makespan, maximum lateness, or weighted tardiness.

The aircraft sequencing and scheduling problems involve determining the assignment of each aircraft (task) to the runway (machine) and the start time of the operation (landing or departure) for the aircraft.

Table 1 Notations used

Notation	Definition
$Ai \in A$ / A={A1; A2......An}	set of aircraft
$Ri \in$ R/ R={R1,R2.......Rn}	set of runway
Ts	minimum separation between aircraft Ai and Ai+1
Ei	earliest landing time
Li	latest landing time
Ti	Target landing time of aircraft
LoadR	Load of runway
Qlength	Queue length of runway
runway-U	runway utilization
THH	The higher threshold
THL	The lower threshold
OLD-list	Overloaded List
ULD-list	Underloaded List
BLD-list	Balanced List
Loadavg	Average Load
NBRR	Number of runways
NBRA	Number of Aircraft

As a result, the following assumptions are taken into account for the aircraft sequencing and scheduling problem:

- A non-preemptive system exists in which a process cannot be halted until it is completed.
- The earliest available time to take off or land at runway end, excluding taxi time, is referred to as "ready time" (i.e., ready times are not necessarily zero).
- The runways are constantly open and reliable.
- At any given moment, only one aircraft is permitted to operate on each runway.
- Separation times, ready times, target times, and deadlines are all dependent on the order in which they are performed and the aircraft type.
- The technology constraints, operation type, aircraft sizes, and penalty weights are all determined and known in advance.
- Each aircraft has a window of time within which it must land or depart, and the final condition requires a fair amount of time or distance between successive landings or departures.

The problem elements and definitions are listed below in Table 1.

Depending on the current load of each runway, the controller decides to start an assignment of aircraft.

A load of the runway at a given time was described by queue length and runway utilization. Queue length denotes the number of aircraft which are waiting to be assigned, we considered Runway-U (Runway Utilization), and Qlength (Queue length) as load information parameters to measure the load of a runway.

The parameters are calculated as follow:

Load (Runway-U)= $\sum_{i=1}^{T} U_i / T$ where: U_i is the value of Runway-U in a previous one second interval, and T is the number of time intervals.

Load (Qlength)= $\sum_{i=1}^{T} Q_i / T$ Where: Q_i is the value of Qlength in a previous one-second interval, and T is the number of time intervals.

The averaged information of Runway-U and Qlength is the load parameters used to describe the load of the runway.

The controller classifies the runways according to their load parameters. It used three states for classification: overloaded, under loaded, and balanced. The first time, it must calculate two threshold values for each load parameter (Runway-U and Q length). The calculation of these thresholds is done as follow:

Calculate load average of each parameter (Runway-U and Qlength) over all runways

$\text{Load}_{avg}(\text{Qlength})= \sum_{i=1}^{NBRR} Load(Qlength)_i / NBR_R$; Where:

• Load_{avg}(Qlength) is the average load (Qlength) over all runways.

• Load $(\text{Qlength})_i$ is the current Qlength of each runway.

$\text{Load}_{avg}(\text{Runway-U}) = \sum_{i=1}^{NBRR} Load(Runway-U)_i / NBR_R$ Where:

• Load_{avg} (Runway-U) is the average load of Runway-U over all runways.

• Load $(\text{Runway-U})_i$ is the current load of Runway-U of each runway.

Calculate the threshold values

The higher and lower threshold values of load parameters are calculated by multiplying the average load of each parameter and a constant value.

THH(Qlength)=H*Load_{avg} (Qlength), THL(Qlength)=L* Load_{avg}(Qlength);

THH(Runway-U)=H*Load_{avg} (Runway-U), THL(Runway-U)=L* Load_{avg}(Runway-U);

• THH is the high threshold

• THL is the low threshold

• H and L are constants and were estimated based on a history realized after several experiments. We found that the best results are obtained for H=1, L=0.6.The next step is to partition the runways for balanced, overloaded and underloaded runways by using the threshold values as follow:

$$\text{Load}=\begin{cases}\text{Balanced} \longrightarrow \text{otherwise}\\ \text{Overloaded} \longrightarrow \text{(Runway-U is high) or (Qlength is high)}\\ \text{Underloaded} \longrightarrow \text{(Runway-U is low) or (Qlength is low)}\\ \text{Idle} \longrightarrow \text{Runway-U} \le 1\%\end{cases}$$

• Overloaded: the runway will be added to the overloaded List, if any of these conditions is true:

1) Runway U is high; if the runway usage is higher than THH (Runway-U), then the runway is overloaded.

2) Qlength is high; if the number of aircraft in the queue of a runway is higher than THH (Qlength), then the runway is classified as an overloaded runway

• Under-loaded: the runway will be added to the under-loaded list if either of these conditions is true:

1) Runway U is low; if the runway usage is lower than THL(Runway-U), then the runway is

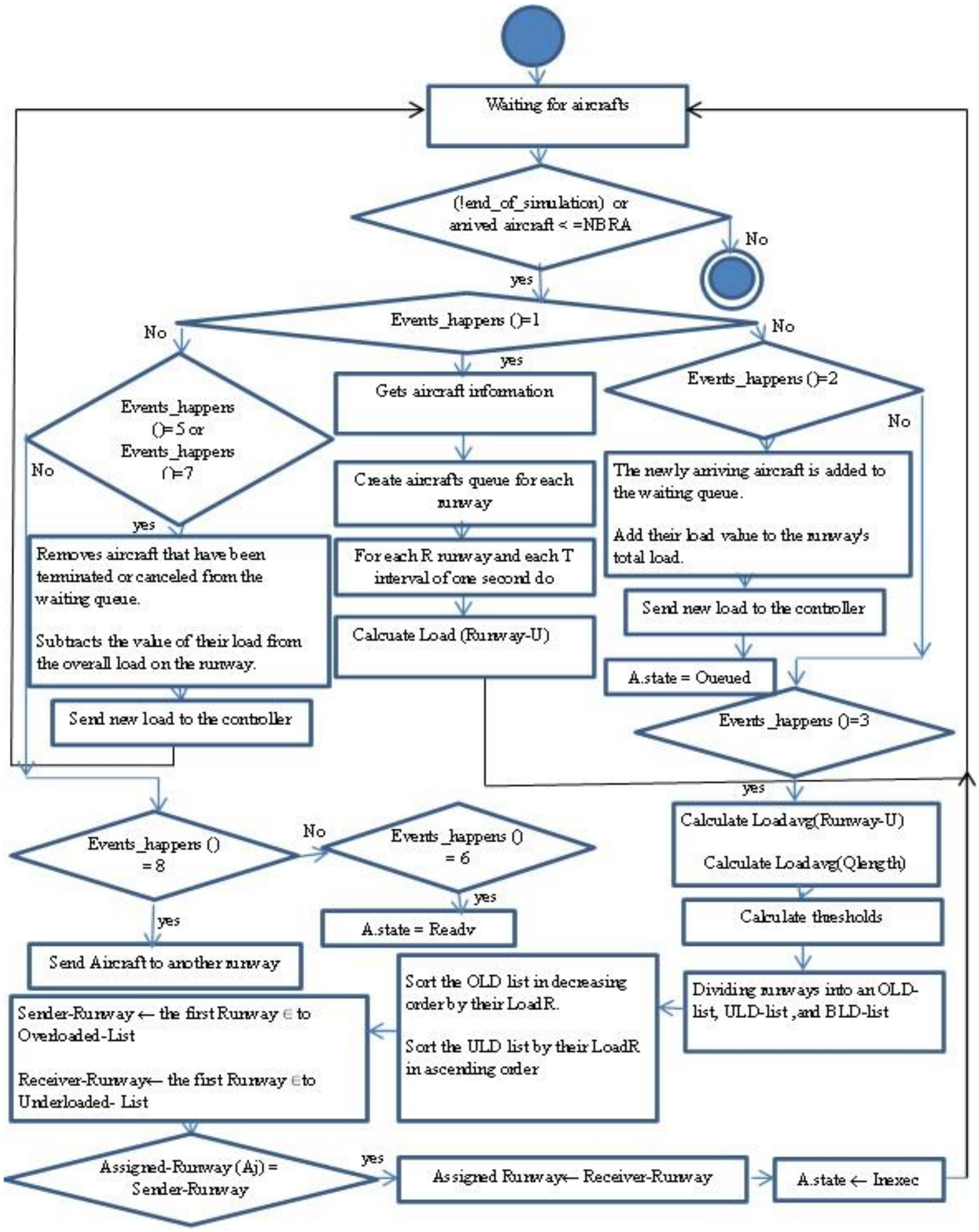


Fig. 1 Flowchart of the proposed algorithm

added to the under-loaded list.

2) Qlength is low; if the number of aircraft in the queue of a runway is lower than THL(Qlength), then the runway is classified in the low state.

• Balanced: the runway is not in the overloaded list and the under-loaded list, and then the runway is in the balanced load state. They are considered more loaded than the low state and less loaded than the high state.

After classifying the runways, in the next step, the controller decides to assign aircraft to Under loaded runways. Fig. 1 shows the flowchart of the proposed algorithm.

There are some specific events which modify the load on multiple runways which can be categorized as follows:

- Any new aircraft is arrived // Aircraft State="Arrived"
- The aircraft has been landed/taken off successfully // Aircraft State="Success"
- Any new runway is added
- Aircraft failure at any runway// Aircraft State="Failed "
- Any existing runway is withdrawal

The aircraft failed due to a runway failure //

Aircraft State=Failed_Runway_Unavailable

- Runway become overloaded

When any of these events happen, the load value is changed and the runway load information is updated. The controller estimates the capability of each runway and estimates its current load based on load information, and it determines the overloaded and the underloaded runways depending on the queue length and runway utilization of every runway. After classifying the runways, in the next step, the controller decides to migrate aircraft from overloaded to under-loaded runways. Finally, the controller decides to assign the aircraft to Receiver-Runway (under-loaded runway). The aircraft States are listed below in Table 2.

3. Experimental results

We choose to compare the suggested algorithm's performance against that of the FCFS algorithm. We note the FCFS (First Come First Served) is the most popular solution to the Aircraft Scheduling problem. So both methods are implemented in Java and they run on Intel I3 Duo 2.00 GHz PC with 4GB of RAM.

3.1 Simulated parameters

To evaluate the performance of the proposed approach, we implemented the proposed algorithm in Java using JDK 1.8. The following parameters are used:

Table 2 Aircraft states

Events_happens	Aircraft State
1	Arrived
2	Ready
3	Queued
4	Inexec
5	Success
6	Failed
7	Canceled
8	Failed_Runway_Unavailable

Runway parameters: these parameters give information about available Runways during the scheduling period such as:

• Number of Runways
• Runway capacity
• Date to send load information from Runways
• Tolerance factor.

Aircraft's parameters: these parameters include:

• The number of aircraft queued at every runway.
• Arrival time, waiting time, submission time, start time, processing time, tardiness time and finish time of each aircraft.
• Aircraft characteristics
• Aircraft priority.

Load index: queue length and runway utilization

• Queue Length denotes the number of waiting aircraft in the runway queue.

3.2 Performance parameters

We are concerned with the following parameters:

// The total time spent by all aircraft waiting to land or takeoff is the system delay, which is linked to efficiency. The maximum delay is linked to equity and refers to the amount of time an aircraft can wait before landing or takeoff.

• Average waiting time: denotes the time that all aircraft spend waiting before starting their landing or takeoff.

Average waiting time=max (scheduling time - processing_time) of all aircraft.

// To arrange landings and departures as close to their target times as possible, minimizing total weighted tardiness is a good objective function.

• The tardiness occurs for aircraft *Aj*, when the completion time of any aircraft *j* is greater than its due time.

// We find the most efficient runway utilization by balancing the load of runways.

• Average Runway usage rate

3.3 Experimental results

The performance of the proposed algorithms mentioned in the previous part is simulated and evaluated in this section. The main goal is to see how effectively the proposed algorithms address the Aircraft Sequencing and Scheduling Problems.

Firstly, we report results obtained from implementing the proposed algorithm involving multiple runways (NBRR=5).

Until all departure and arrival aircraft take off or land, the simulation continues. Every 5 minutes, each arriving aircraft's runway assignment is optimized. The time window is set for the next 1 hour.

Table 3 presents our computational results for the proposed algorithm and its improved variants over the classical FCFS algorithm. The first column specifies a time window of 1 hour. Column 2, 3 and 4 represents the number of flights (total, departures and arrivals respectively) in a time window. The delay time is also shown in the table for the traditional FCFS algorithm and for the proposed algorithm.

Table 3 Computational results

Time window	number of flights			delay time	
	Total	departures	arrivals	FCFS	proposed algorithm
6:00-7:00	4	2	2	04.23 min	1.02 min
7:00-8:00	6	4	2	5 min	0.25 min
8:00-9:00	2	1	1	3.04 min	1.22 min
9:00-10:00	7	5	2	10 min	2.54 min
10:00-11:00	3	1	2	3 min	1.25 min
11:00-12:00	10	6	4	12 min	05 min
12:00-13:00	5	2	3	7.33 min	04 min
13:00-14:00	8	4	4	13 min	08 min
14:00-15:00	4	3	1	6.78 min	3.4 min

Table 4 Queue length for each runway using FCFS algorithm with 2000 aircraft

Runway ID	Runway Name	Queue Size
25	Runway_4	0
17	Runway_2	1999
9	Runway_0	1
13	Runway_1	0
21	Runway_3	0

Table 5 Queue length for each runway using proposed algorithm with 2000 aircraft

Runway ID	Runway Name	Queue Size
25	Runway_4	539
17	Runway_2	72
9	Runway_0	1381
13	Runway_1	0
21	Runway_3	8

There is an increase in Total delays (13 min) in the time window (13h-14h) for the traditional FCFS algorithm because in the FCFS algorithm, if the first aircraft in the queue fails, the remaining aircraft cannot be scheduled even though the requested runways are available. In the proposed algorithm, we note that there is a minimization of the total delay time in the time window (13h-14h) from 13 min to 8 min, the reason for this minimization is that the proposed algorithm has the possibility to change aircraft priorities in real-time so that airports can manage all landing requests during the period of risk and disaster.

3.3.1 Queue length

The findings revealed that FCFS has the problem of rarely providing ideal sequences in terms of runway throughput or average delay in congested airports.

From Fig. 2, we can perceive that the queue size of runway as shown in Table 4. ID 17 is 500 with 500 aircraft, 999 with 1000 aircraft, 1498 with 1500 aircraft, and 1999 with 2000 aircraft, while runway ID 25,13, and 21 are idle. The proposed algorithm balances the load between runways at the time of scheduling. It shows that the proposed algorithm is better performed

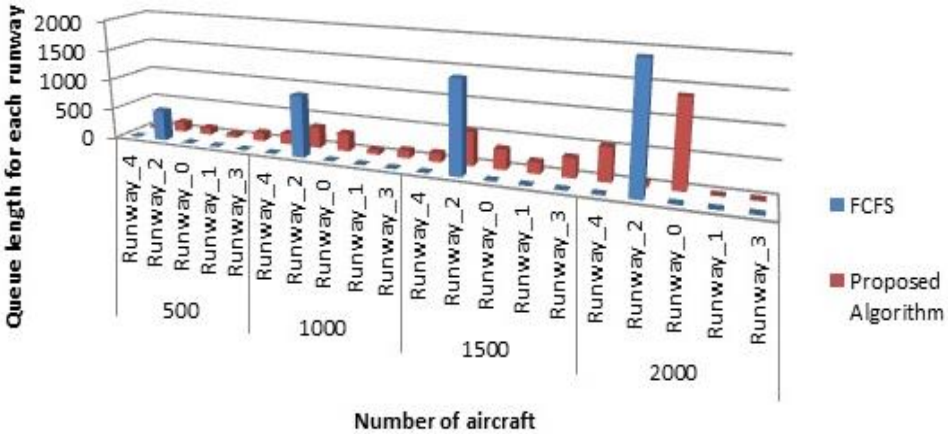


Fig. 2 Comparison of Queue length for each runway of proposed algorithm and FCFS algorithm

compared to the FCFS algorithm, also because load balancing is used to make sure that none of the existing runways is idle while others are being utilized. The load balancing effects are caused by under-loaded runways. In the proposed algorithm there is an increase in the queue size of runway 25 from 0 (FCFS algorithm) to 539 (after). Runway 9 from 1 to 1381 and runway 21 from 0 to 8 with 2000 aircraft. With 1500 aircraft, we can perceive that there is an increase in the queue size of runway ID 25 from 0 to 138, runway ID 13 from 0 to 185, and runway ID 21 from 0 to 311. Also, there is an increase in the queue size for runway ID 25, from 0 to 177. Runway ID 13, from 0 to 70. Runway ID 25, from 0 to 125 with 1000 aircraft. In the proposed algorithm, the different utilizations of the participating runways are balanced.

3.3.2 Average waiting time

We focused on aircraft in the waiting queue and compared them to aircraft that were already flying. The number of runways is assumed to be five, and the number of aircraft is assumed to be 2000.

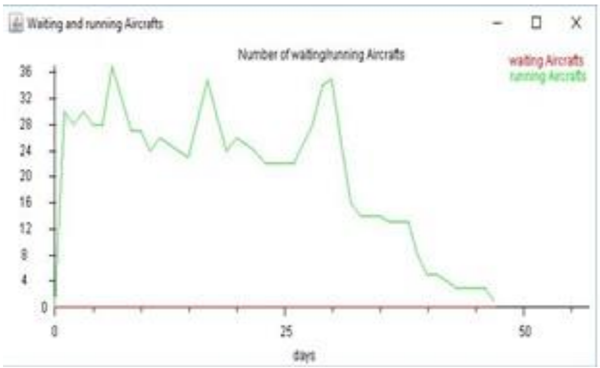
In the FCFS algorithm, if there are some aircraft already in a departure queue, the aircraft will have to wait. But in the proposed algorithm, when the first aircraft in the queue cannot be scheduled directly, the proposed algorithm estimates the earliest probable starting time for the first aircraft using the processing time calculated for flying aircraft. Then, it makes a reservation to run the aircraft at this pre-estimated time. Next, it examines the queue of waiting jobs and directly schedules every aircraft not intervening with the reservation of the first aircraft.

In Fig. 3, The horizontal axis represents time (units of days) while the vertical axis indicates the number of aircraft. The red curve shows the number of waiting aircraft, which says that an aircraft is in the waiting aircraft queue, and the green curve shows a number of running aircraft, which says that an aircraft is flying. FCFS is not able to schedule aircraft easily, generating considerable waiting time for aircraft. For 2000 aircraft and with 5 runways, the proposed algorithm is capable of higher runway utilization and there is no waiting aircraft in the queue throughout the time.

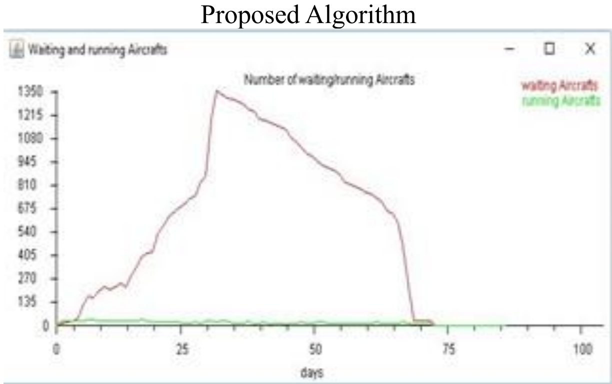
3.3.3 Tardiness

The goal is to reduce overall weighted Tardiness as much as possible.

When minimizing aircraft tardiness, we take into account the maximum consecutive delay.



Proposed Algorithm



FCFS algorithm

Fig. 3 Comparison of flying Aircrafts and waiting Aircrafts with 2000 aircrafts

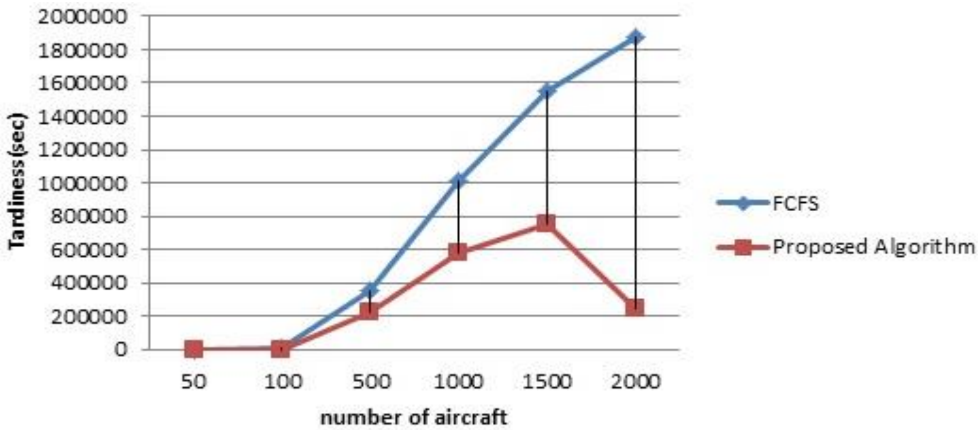


Fig. 4 Comparison of average tardiness between FCFS and the proposed algorithm.

From Fig. 4, we note the maximum delay is 753742 seconds for 1500 aircraft using the proposed algorithm. In the FCFS (first come, first served) algorithm, the maximum delay is 1870139 seconds for 2000 aircraft. It is apparent that in FCFS, some of the aircraft are put into a waiting state because there is a rise in the maximum delay compared to the proposed algorithm. The proposed algorithm eliminates this probability. However, some of the aircraft have to wait, which

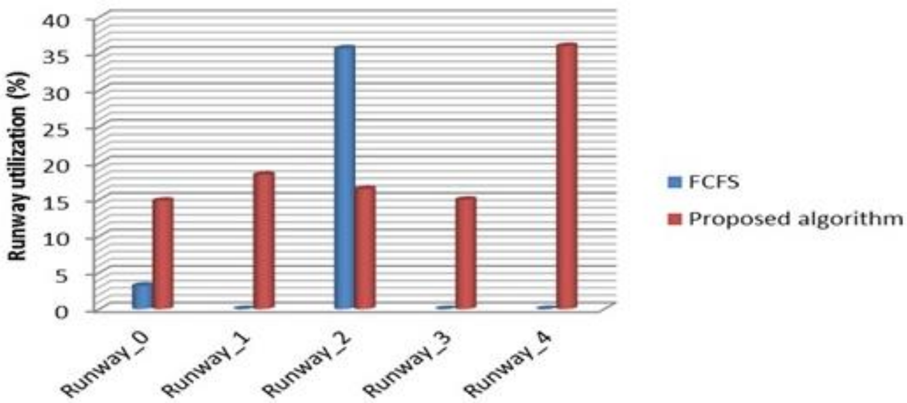


Fig. 5 Comparison of Runway Utilization (%) between FCFS and the proposed algorithm with 1500 aircrafts

is one of the advantages of the proposed algorithm.

3.3.4 Average runway usage rate

Efficient runway usage in the airport is a speedy and feasible solution as compared to airport development and runway construction.

Examination of the runway utilization (Fig. 5) showed that FCFS did not perform well. It shows that runway-2 which is having the highest capacity is over-utilized, while runways 1, 3, and 4 are idle in the FCFS algorithm. The reason behind the improvement is even utilization of all runways which is achieved because the proposed algorithm balances the load between runways at the time of scheduling. It shows that the proposed algorithm is better performed compared to the traditional algorithm, also because load balancing is used to make sure that none of the existing runways is idle while others are being utilized. The load balancing effects are caused by under-loaded runways. In the proposed algorithm there is an increase of utilization of runway-0 from 3,19% to 14,83% (after) and runway-1 from 0% to 18,4% and runway-3 from 0% to 14,98 % and runway-4 from 0% to 36,03 %.

5. Conclusions

In In this paper, we have examined aircraft sequencing and scheduling problems on multiple runways. Compared with the FCFS algorithm, the main benefits of the proposed algorithms in this paper are as follows:

- The simplicity of the proposed algorithm
- The proposed algorithm considers the heterogeneity of runways.
- We utilized a dynamic scheduler of aircraft since it was always better in congested airports.
- The runways studied in this paper are multiple parallel runways, which accords with the development trend of airports nowadays.
- Hard restrictions are considered in the article, such as the minimum separation time between aircraft, the prohibition of aircraft assignments before their scheduled time, and the prohibition of two specific aircraft being assigned at the same time.
- The proposed algorithm can produce a practical aircraft sequencing solution.

• The aircraft will be reallocated to other runways when a runway fails.

Results show that when compared with FCFS scheduling, the proposed algorithms are effective in achieving an efficient runway throughput. In addition, the algorithms are capable of finding solutions that perform well in terms of minimizing average waiting time, minimizing average tardiness, and maximizing runway utilization. However, due to the complexity of aircraft sequencing and scheduling problems on multiple runways, it is necessary to make adjustments based on real-time data. As a result, future research and development of the algorithm to accommodate the current complicated scheduling environment and achieve a mutually acceptable result. The proposed method will be tested under real-world operational data to see how effective it is. These will be the focus of future research.

References

Balakrishnan, H. and Chandran, B. (2006), “Scheduling aircraft landings under constrained position shifting”, *IAA Guidance, Navigation, and Control Conference and Exhibit*, Key-stone,

Balakrishnan, H. and Chandran, B. (2010), “Algorithms for scheduling runway operations under constrained position shifting”, *Oper. Res.*, **5**(6), 1650-1665. https://doi.org/10.1287/opre.1100.0869.

Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M. and Abramson, M. (2000), “Scheduling aircraft landings: The static case”, *Transp. Sci.*, **34**(2), 180-197. https://doi.org/10.1287/trsc.34.2.180.12302.

Bo, S., Ming, W. and Binbin, J. (2021), “Optimal model for the aircraft arrival and departure scheduling problem with fuzzy runway incursion time”, *J. Math. Biosci. Eng.*, **18**(5), 6724-6738. https://doi.org/10.3934/mbe.2021334.

Chandran, B. and Balakrishnan, H. (2007), “Dynamic programming algorithm for robust runway scheduling”, *2007 American Control Conference*, July.

Fischetti, M. and Salvagnin, D. (2010), “An in-out approach to disjunctive optimization”, *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, Berlin, Heidelberg, June.

Pohl, M. Kolisch, R. and Schiffer, M. (2021), “Runway scheduling during winter operations”, *Omega*, **102**, 102325. https://doi.org/10.1016/j.omega.2020.102325.

Rathinam, S., Montoya, J. and Jung, Y. (2008), “An optimization model for reducing aircraft taxi times at the Dallas Fort Worth International Airport”, *26th International Congress of the Aeronautical Sciences*, Anchorage, 14-19. https://doi.org/10.1177/0954410011433238.

Venkatakrishnan, C.S., Barnett, A. and Odoni, A.R. (1993), “Landings at Logan Airport: Describing and Increasing Airport Capacity”, *Transp. Sci.*, **27**(3), 211-227. https://doi.org/10.1287/trsc.27.3.211.

EC