

Experiment No. 8

Title: Implement Airline Scheduling Algorithm using

python. Aim:

To explore, understand and implement the Airline Scheduling algorithm in python programming language, aiming to find an optimal schedule which can select all the flights which are possible in the provided time slots.

Objectives:

1. Implement the Airline Scheduling algorithm in python programming language.
2. Demonstrate the effectiveness and efficiency of the algorithm in finding an optimal schedule of non-overlapping flights.
3. To consider constraints such as departure time, arrival time and maintenance time.

Problem Statement:

Provided a list of flights along with the flight number, departure time, arrival time and maintenance time. The goal is to find an optimal schedule for the airlines such that the maximum number of non-overlapping flights are scheduled along with considering constraints such as maintenance time..

Software/s: Python (version 3.x)

Theory:

Airline Scheduling Problem:

Airline scheduling algorithms are essential for optimizing the allocation of resources such as aircraft, crews, and airport facilities to efficiently operate flights while meeting various constraints and objectives. The goal of airline scheduling algorithms is to create feasible schedules that maximize revenue, minimize costs, and ensure safety and customer satisfaction.

Airline schedules must adhere to various constraints, including aircraft availability, crew scheduling regulations, airport slot allocations, maintenance requirements, and airspace restrictions. Airlines typically aim

to optimize one or more objective functions, such as maximizing revenue by maximizing the number of passengers carried, minimizing costs by reducing turnaround times, or ensuring on-time performance to enhance customer satisfaction. Airline scheduling algorithms employ various optimization techniques, including greedy algorithms, dynamic

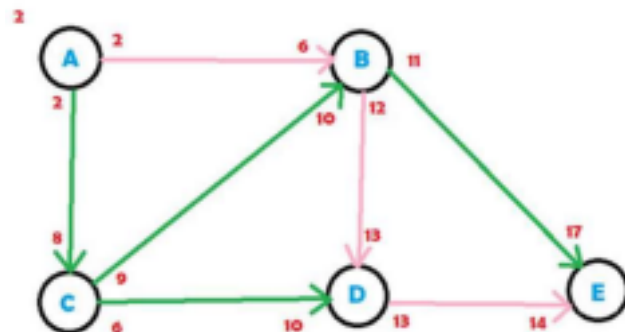
Department of Artificial Intelligence and Data Science



programming, integer programming, genetic algorithms, and metaheuristic approaches like simulated annealing and tabu search. Airlines need to adapt schedules in real-time to accommodate disruptions such as weather events, mechanical failures, crew shortages, and air traffic congestion while minimizing the impact on operations. Airline scheduling algorithms are integrated with other operational systems such as flight planning, crew management, revenue management, and customer service to ensure the coherence and effectiveness of the overall operation.

Airline scheduling algorithms are complex optimization problems that require careful consideration of resource allocation, constraints, objectives, optimization techniques, schedule robustness, and technological integration to achieve efficient and effective scheduling outcomes.

Flight i	Origin $\tau(i)$	Destination $\tau'(i)$
1	Airport a_1 7.00	Airport a_3 9.00
2	Airport a_1 10.00	Airport a_3 12.15
3	Airport a_2 7.30	Airport a_4 8.30
4	Airport a_2 9.00	Airport a_4 12.00
5	Airport a_2 12.00	Airport a_5 14.15



Algorithm:

1. Define a Flight class with attributes for flight number, departure time, arrival

Department of Artificial Intelligence and Data Science



time, and maintenance time.

2. Implement a function to schedule flights considering maintenance time constraints.
 - i. Sort the list of flights based on their arrival times.
 - ii. Initialize an empty list schedule to store scheduled flights.
 - iii. Initialize a variable current_time to track the current time.
 - iv. Iterate through each flight in the sorted list:
 - a. Check if there's enough time for maintenance and then departure: b. If the departure time of the flight is greater than or equal to the current time plus maintenance time:
 - c. Add the flight to the schedule.
 - d. Update the current time to the arrival time of the flight.
 - v. Return the list of scheduled flights.
3. Implement a function to obtain flight details from the user.
4. Initialize an empty list flights to store flight objects.
5. Prompt the user to enter flight number, departure time, arrival time, and maintenance time.
6. Schedule the flights using the Airline Scheduling Algorithm.
7. Display the scheduled flights with their details.

Pseudocode:

```
class Flight:
```

```
// Flight class definition...
```

```
function convert_to_minutes(time_str):
```

```
    hours, minutes = split time_str by space
```

```
    return hours * 60 + minutes
```

```
function schedule_flights(flights):
```

```
    sort flights by arrival_time
```

```
    initialize schedule as an empty list
```

```
    current_time = 0
```

```
    for each flight in flights:
```

```
        if flight.departure_time >= current_time +
```

```
        flight.maintenance_time: append flight to schedule
```

```
        current_time = flight.arrival_time
```

```
    return schedule
```

```
function get_flights_from_user():
```

```
    initialize flights as an empty list
```

```
    while True:
```

```
flight_number = input("Enter flight number (or 'done' to finish): ")
if flight_number.lower() == 'done':
    break
departure_time_str = input
departure_time =
convert_to_minutes(departure_time_str) arrival_time_str
= input
arrival_time = convert_to_minutes(arrival_time_str)
maintenance_time = input
add current flight to our flights list
return flights

get flights from user
call the schedule_flights function
print the result
```

Conclusion:

Department of Artificial Intelligence and Data Science