Airline Crew Scheduling: Backtracking Solution

NP-Hard Problem Analysis and Profiling

Ayush Singh
Roll No: 2301201086

November 17, 2025

Introduction

The airline crew scheduling problem is a classic NP-hard problem where the goal is to assign a set of flights to available crew members such that all schedule and rest constraints are satisfied. This project implements a backtracking algorithm for small flight datasets, demonstrates complexity through profiling, and provides informative visualizations.

Problem Definition

Given a set of flights defined as $(\text{FlightID}, \text{Start Time}, \text{End Time})$ and a set of crew members, the problem is to assign crew to flights such that:

- No crew member has overlapping flights,
- Each crew member gets a minimum required rest between consecutive flights,
- All flights are assigned to some crew member.

Algorithm Approach

A backtracking algorithm systematically assigns flights to crew members and recursively checks for constraint satisfaction:

- **No Overlap:** Flights assigned to the same crew do not overlap in time.
- **Minimum Rest:** Consecutive flights assigned to one crew have at least 1 hour rest.

The algorithm explores all valid assignments in $\mathcal{O}(k \times 2^n)$ where n is the number of flights and k is the number of crew.

Code Snippet (Python)

```
def is_overlapping(f1, f2):
    _, s1, e1 = f1
    _, s2, e2 = f2
    return not (e1 <= s2 or e2 <= s1)

def has_minimum_rest(f1, f2, min_rest):
    _, s1, e1 = f1
    _, s2, e2 = f2
    if e1 <= s2:
        return (s2 - e1) >= min_rest
    elif e2 <= s1:
        return (s1 - e2) >= min_rest
    else:
        return False
```

Profiling and Scalability

The algorithm is tested with increasing flight numbers to monitor time and recursive calls. Profiling code shows exponential growth in resource usage.

```
Flights:  4 | Time: 0.000000s | Calls:      5 |  Found
Flights:  5 | Time: 0.000000s | Calls:      6 |  Found
Flights:  6 | Time: 0.000000s | Calls:      7 |  Found
...
Flights: 10 | Time: 0.002797s | Calls:     35 |  Found
```

Memory profiling:

> Peak memory: 112.09 MiB, increment: 0.18 MiB

Visualization

The Gantt chart below visualizes the flight assignment to each crew member.

Complexity Analysis & Insights

- **Time Complexity:** The backtracking algorithm's worst-case time complexity is exponential ($\mathcal{O}(k \times 2^n)$).
- **NP-Hardness:** Constraint satisfaction and combinatorial search space make the problem intractable at large scale.
- Backtracking is practical only for small instances (10-15 flights).

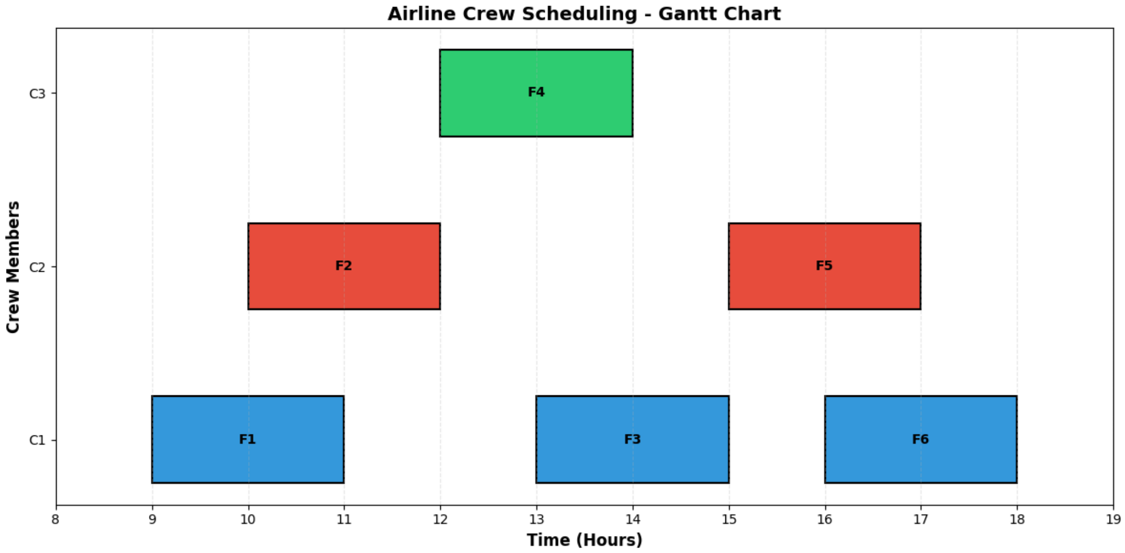

Figure 1: Flight Schedule Assignment (Gantt Chart)

Improvements & Limitations

Improvements:

- Use Integer Programming or specialized constraint solvers for large real-world problems.
- Apply heuristics and pruning to reduce computation.

Limitations:

- Exponential growth in computation.
- Not directly practical for industrial-size airline schedules.

References

1. Airline Crew Scheduling Problem – Wikipedia
2. Matplotlib Documentation
3. Memory Profiler for Python