Efficient Software-Based Allocation of Commercial Flight Seats

Shiven S. Patel
pate1681@purdue.edu
Department of Computer and Information Technology
Purdue University
West Lafayette, Indiana, USA

ABSTRACT

A software program is developed using the C language to explore how aircraft seats are allocated to passengers on commercial flights. A handful of algorithms are used to process and operate on data, which is ultimately used to make seating decisions.

ACM Reference Format:
Shiven S. Patel. 2024. Efficient Software-Based Allocation of Commercial Flight Seats. In *Proceedings of Spring 2024 CNIT 315 (CNIT 315).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

According to the Bureau of Transportation Statistics, an agency of the United States Department of Transportation, U.S. airlines carried over 853 million passengers in 2022, a full 30 percent increase over the previous year.[7] With commercial airline traffic steadily growing, and demand for leisure and business travel continuing to recover from pre-COVID-19 numbers, airlines face a need to efficiently seat passengers.

1.1 Motivation

At scale, efficiently seating passengers requires well-built and reliable software systems that can analyze variables and create a well-balanced seating chart that aligns the interests of passengers and the airline. A key element of these software systems is the algorithm that decides who gets what seat. The motivation behind this software project included a desire to understand these algorithms, understand how airlines allocate seats, and more broadly, understand how these software systems work.

2 LITERATURE REVIEW

A joint study conducted in 2020 by researchers from Sejong University and Daejin University in the Republic of Korea explored a similar idea. The study, conducted by Young Dae Ko, Sung Il Kwag, and Yonghui Oh, was titled *An efficient airline seat reallocation algorithm considering customer dissatisfaction* and it explored a potential algorithm for assigning plane seats to passengers, with the objective being to reduce passenger dissatisfaction as much as possible. Of particular interest to this project were the seat features the authors used to inform the design of their algorithm. Specifically, the authors relied on the assumption that "In general, the customer tends to prefer a seat in the front row rather than the rear row, because it is more convenient to get on and off the aircraft. In addition, there is a tendency to prefer an aisle or a window seats rather than the intermediate seats to secure the private spaces". [3] From this assumption, seating position farther to the front of the aircraft was used to guide the development of the program discussed in this paper, for the sake of simplicity. A potential future improvement to the program would include support for specific seat positions (i.e. window, middle, aisle for '3x3' or similar seat layouts).

3 OBJECTIVES

Following a thorough literature review, a set of objectives for the application were compiled. They are as follows:

(1) Seats shall be assigned based on specific parameters pertaining to the passenger. These parameters may include military status, airline rewards status, class of cabin seat, etc. (a more detailed breakdown will come in Section 4.3). The parameters will calculate a score for each passenger, which will be used to assign seats.
(2) Sorting shall be used in addition to the scoring algorithm. Sorting should take the calculated score and use it to sort the passengers in the cabin accordingly. A priority for the sorting algorithm will be efficiency.
(3) The program shall support various aircraft from a variety of manufacturers, most notably U.S.-based Boeing and Europe-based Airbus. Boeing and Airbus are the two largest manufacturers of commercial aircraft by volume of deliveries as of Fiscal Year 2021. [5]

4 APPROACHES AND METHODS

4.1 Taking Passenger Data Input

The first step undertaken by the program is to read the input file it must be provided. The input file should be in the `.txt` file format, which each line containing information about a single passenger. Each line is to be formatted as follows:

```
firstName, lastName, over65, rewardsTier,
milStatus, cabinClass
```

The `fgets()` function is called to retrive each line as follows: `fgets(fgets(new, 120, cabin))`.

4.2 Data Ingestion and Processing

Once the function has access to the line, it uses the `strtok()` function to extract the data into local variables. For each line, a separate function is called to create and dynamically allocate a struct using `malloc()`, with the values from the text file being populated into

CNIT 315, April 22–30, 2024, West Lafayette, IN

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

the corresponding variables in the struct. The struct is defined as follows:

```
typedef struct Passenger {
    char firstName[50];
    char lastName[50];
    int passengerID;
    int over65;
    int rewardsTier;
    int milStatus;
    int cabinClass;
    double priority;
} Passenger;
```

Each struct is then placed into an array. The array's size represents the total number of seats on the aircraft.

4.3 Operating on the Data

A function in the program, called `calculatePriority()`, takes the filled struct in and calculates a weighted score for each passenger. The weights were chosen based on criteria airlines use when boarding passengers. The weights are as follows:

Cabin Class: 40 percent Cabin class in this case refers to the type of economy ticket a passenger has, with the assumption that the airline lets First and Business class passengers choose their seats. Premium Economy gets a score of 5, Standard Economy gets a 3, and Basic Economy gets a 1.

Rewards Tier: 25 percent Rewards programs are of great importance to airlines. In fact, during the COVID-19 pandemic, loan applications to creditors from airlines revealed that the airline frequent flier programs were worth more than the airlines themselves.[1] As such, an arbitrary 5 point scale was used, describing rewards tiers as follows: 1 - Bronze, 2 - Silver, 3 - Gold, 4 - Platinum, 5 - Diamond.

Military Status: 20 percent Airlines often offer discounts and other benefits to members of their airline's home nation's armed forces. This is especially true in the United States, where nearly every major airline offers military discounts of some kind. In this scale, active duty military gets a 5, veterans get a 3, and all other get a 0.

Senior Citizen Status: 15 percent Airlines have an interest in ensuring the often uncomfortable experience of flying is as comfortable as possible for senior citizen passengers, who may have more issues with flying. A score of 5 is given to passengers over the age of 65, and 0 for passengers under 65.

A simple insertion sort then took the array and sorted it in descending order, based on the calculated metric:

```
for (int i = 1; i < totalPassengers; i++) {
    Passenger *p = flatMap[i];
    int j = i - 1;

    while (j >= 0 && flatMap[j]->priority < p->priority) {
        flatMap[j + 1] = flatMap[j];
        j = j - 1;
    }
    flatMap[j + 1] = p;
}
```

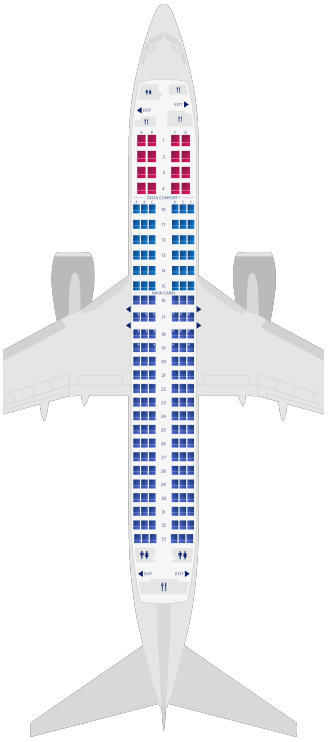

Figure 1: Delta Airlines Boeing 737-800 Layout [2]

seating_chart.txt

```
Row 1:
    Seat 1A: Wilson, David (ID: 5)
    Seat 1B: Taylor, Robert (ID: 7)
    Seat 1C: Thomas, William (ID: 9)
    Seat 1D: Young, Amanda (ID: 12)
    Seat 1E: Rodriguez, James (ID: 13)
    Seat 1F: Scott, Mary (ID: 16)
Row 2:
    Seat 2A: Hall, Laura (ID: 18)
    Seat 2B: Green, Brittany (ID: 20)
    Seat 2C: Hill, Nicholas (ID: 23)
    Seat 2D: Carter, Joshua (ID: 25)
    Seat 2E: Moore, Andrew (ID: 27)
    Seat 2F: White, Stephanie (ID: 30)
Row 3:
    Seat 3A: Roberts, Samuel (ID: 33)
    Seat 3B: Hall, Megan (ID: 36)
    Seat 3C: Wood, Victoria (ID: 38)
    Seat 3D: Scott, Heather (ID: 42)
    Seat 3E: Rivera, Michelle (ID: 44)
    Seat 3F: Nguyen, Christina (ID: 46)
```

Figure 2: Example Seating Chart

4.4 Exporting Seating Chart

Generally, similar size and category of aircraft may share seat layouts. For example, Figure 1 shows a sample layout of the seating chart for U.S.-based Delta Airlines' Boeing 737-800 aircraft. As shown in Figure 1, the main cabin has approximately 30-33 rows of seats, with 6 seats in each row grouped into two groups of three. Therefore, a series of nested `for` loops can be used to iterate through the array and generate a seating chart. Figure 2 shows three rows from an example seating chart.

5 RESULT AND ANALYSIS

Overall, the program functioned as desired, fulfilling all of the originally outlined objectives. Additionally, the program utilized input and output files, which represented distributed elements.

5.1 Efficiency Analysis

Many choices made in the design of the program promote efficiency. Fundamentally, by using a low-level language like C, the program is efficient by nature. Additionally, using `.txt` files promotes efficiency due to their simple nature and low overhead. The program also has a significant utilization of dynamically allocated memory. All of the `Passenger` structs are dynamically allocated, which is more efficient than static allocation. [6] The sort utilized by the program, insertion sort, is also efficient and simple by design. [4]

5.2 Challenges

The most significant challenge faced during the development of the application was the inclusion of the API. Originally, the plan for API implementation involved using the FlightLookup API. Under this design, the user would enter a flight number, and the program would call the API to retrieve information about the flight, specifically the aircraft model. The program would then check that model number against the program's list of supported models, generating an appropriate seating chart if the aircraft's layout was supported. Unfortunately, the library used in C to parse the XML output from the API, `libxml2`, was failing to retrieve the correct element in the data, despite the XPath expression being verified and correct.

5.3 Future Improvements

While the program did successfully meet all of the expressed objectives, there are some improvements that can be made. They are as follows:

(1) The program should support more complex seat layouts, including the inclusion of first and business class seats.
(2) The program should account for aisle, window, and middle seats, as well as passenger preferences for each. Additional attention may also be need to be given to more unique layouts, such as the Boeing 777-300ER, which has a 3x4x3 row layout cabin.
(3) The program should account for groups of passengers who travel together, specifically families.
(4) The program should incorporate a more advanced scoring system, with additional variables and greater complexity.

6 CONCLUSION

Overall, the application developed met the expressed objectives in an efficient and user-friendly manner. While there is still a host of improvements that may be made to the program to enhance capability, function, and usability, the program is successful in its original goal serving as a practical means of understanding how airlines use software systems to allocate flight seats to passengers. Airlines are often looking to improve passenger experience, and software systems are critical in helping airlines achieve this goal.

REFERENCES

[1] Bushey, Claire. 2020. US airlines reveal profitability of frequent flyer programmes. https://www.ft.com/content/1bb94ed9-90de-4f15-aee0-3bf390b0f85e
[2] Delta Air Lines, Inc. 2024. Boeing 737-800 (738). https://www.delta.com/us/en/aircraft/boeing/737-800
[3] Young Dae Ko, Sung Il Kwag, and Yonghui Oh. 2020. An efficient airline seat reallocation algorithm considering customer dissatisfaction. *Journal of Air Transport Management* 85 (2020), 101792. https://doi.org/10.1016/j.jairtraman.2020.101792
[4] Olawanle, Joel. 2023. Insertion Sort Algorithm - Most Asked Questions About Insertion Sort. https://www.freecodecamp.org/news/most-asked-questions-about-insertion-sort-algorithm/#:~:text=Insertion%20sort%20is%20a%20simple,relative%20order%20of%20equal%20elements.
[5] Statista Research Department. 2024. Key figures of the four largest aircraft manufacturers worldwide in FY 2021. https://www.statista.com/statistics/269920/key-figures-of-the-four-largest-aircraft-manufacturers/
[6] Thakur, Shreeya. 2023. Differences Between Static And Dynamic Memory Allocation (Comparison Chart). https://unstop.com/blog/difference-between-static-and-dynamic-memory-allocation
[7] U.S. Department Of Transportation. 2023. Full Year 2022 U.S. Airline Traffic Data | Bureau of Transportation Statistics. https://www.bts.gov/newsroom/full-year-2022-us-airline-traffic-data#:~:text=For%20the%20full%20year%202022,and%20388%20million%20in%202020.