

CSC111 Project Proposal: FLIGHT SEARCH ENGINE

Andrew Andrew, Nagata Parama Aptana, Khoi Minh Bui, Melissa Cecilia

Sunday, March 5, 2023

Problem Description and Research Question

Planning a trip abroad can often be stressful, and is often worse when the destination is thousands of miles away. Securing a plane ticket is among the most crucial step in planning a trip. Nowadays, people would open Google, input several details such as time, origin city, destination, class, etc.. Afterwards, a list of possible flights would be displayed.

This process might sound simple, but how does this exactly work behind the scene? Through this project, we would like to find out what answer/algorithm lies behind this search engine.

Our research question, in its simplest form, is the following:

Given a set of plane flights with the following properties:

- flightID: The ID of this flight
- startingAirport: Three-character IATA airport code for the initial location
- destinationAirport: Three-character IATA airport code for the arrival location
- departureTime: The time (YYYY-MM-DD HH:MM in GMT+0) this flight departs departs from startingAirport
- arrivalTime: The time (YYYY-MM-DD HH:MM in GMT+0) this flight arrives at destinationAirport
- totalFare: the price of the ticket (in USD) including taxes and other fees

We are given a query with the following arguments/filters

- startingAirport: Three-character IATA airport code for the initial location
- destinationAirport: Three-character IATA airport code for the arrival location
- departureDate: The time (YYYY-MM-DD) searcher wanted to depart
- order_by: True if searcher would like to sort by cheapest price and False if searcher would like to sort by shortest flight duration.

Main Task:

We are tasked to **find the list of feasible flights that meets the searcher's requirement.**

Additional Task:

If several methods are found, we would like to know each method's strength: time-complexity, memory-complexity, updatability (ability to add/delete new flights efficiently), and other conditions.

Several items/properties that can be considered are

- Maximum layover time
- Maximum number of layovers

- Maximum duration of the whole flight
- Requirements of documents/fees, such as VISA, transit-fee, etc.

Computational Plan

The central theme of our research are the flight routes, which will be represented as edges with the attributes given above. The vertices of our graphs will be the airports. However, the graph will be little different from the ones discussed in class. Edges are directed, making this graph a directed graph. Moreover, there can be more than one edges that connects two airports (i.e. when there are multiple flights between the two airports), making this graph a multi-graph.

First, we will need to build the graph of our flight network. We add all the available airport data as vertices into our graph, then add all flight data as edges. For efficiency purposes, we will store the edges in increasing flight-time.

```
class Graph:
    """A graph represent airports

    Instance Attributes:
        - airports: A collection of the airport verticies contained in this graph
    """

class Airport:
    """A node in the graph that represents a single airport.

    Instance Attributes:
        - ID: Three-character IATA airport-code
        - city: The location of the airport
        - flights: A list of flight edges that contain each flight's ID
    """

class Flight:
    """An edge in the graph that represents a flight between two airports.

    Instance Attributes:
        - flightID: The ID of this flight
        - startingAirport: Three-character IATA airport code for the initial location
        - destinationAirport: Three-character IATA airport code for the arrival
            location
        - departureTime: The time (YYYY-MM-DD HH:MM in GMT+0) this flight departs from
            startingAirport
        - arrivalTime: The time (YYYY-MM-DD HH:MM in GMT+0) this flight arrives at
            destinationAirport
        - totalFare: the price of the ticket (in USD) including taxes and other fees
    """
```

For every query, we will use two approaches

1. *Naive Traversal*

We will do standard graph traversal, traversing certain amount of edges depending on the date,

maximum layover time, maximum number of layovers, and other variables that allows this traversal be computationally feasible. Traditionally, the maximum number of layover is 3, which greatly reduces the complexity of this traversal algorithm.

This approach does not rely on precomputing anything, but is using the available data immediately. This allows easy modification on the data, such as insertion or deletion of new flights.

2. *Pruned Landmark Labeling*

This is an optimised algorithm to find the k-shortest optimum path between two airports depending on the parameter set by the user: lowest price or least travel time (Akiba et al., 2015). The rough idea of the algorithm is to precompute a special data structure that prevents the same path to be calculated twice. Then, we send a query consisting of origin, destination, and index to the data structure to get the k-th topmost results.

For our dataset, we found the Airfares in New Zealand data that contains flights with their origin, destination, departure and arrival time, as well as their ticket prices (Ngo, 2020).

Travel Date	Dep. airport	Dep. time	Arr. airport	Arr. time	Duration	Direct	Transit	Baggage	Airline	Airfare(NZ\$)
19/09/2019	AKL	1:35 PM	CHC	3:00 PM	1h 25m	(Direct)	N/A	N/A	Jetstar	111
19/09/2019	AKL	3:55 PM	CHC	5:20 PM	1h 25m	(Direct)	N/A	N/A	Jetstar	111
19/09/2019	AKL	11:40 AM	CHC	1:05 PM	1h 25m	(Direct)	N/A	N/A	Jetstar	132
19/09/2019	AKL	8:00 PM	CHC	9:25 PM	1h 25m	(Direct)	N/A	N/A	Jetstar	132
19/09/2019	AKL	9:00 AM	CHC	10:25 AM	1h 25m	(Direct)	N/A	N/A	Air New Zealand	133

References

- AI, S. (2022, October 2). Flight fare prediction-machine learning project. Medium. Retrieved March 5, 2023, from <https://medium.com/@skillcate/flight-fare-prediction-machine-learning-project-bc7363e6d9eb>
- Akiba, T., Hayashi, T., Nori, N., Iwata, Y., & Yoshida, Y. (n.d.). Efficient top-k shortest-path distance queries on large networks by pruned landmark labeling. Proceedings of the AAAI Conference on Artificial Intelligence. Retrieved March 5, 2023, from <https://ojs.aaai.org/index.php/AAAI/article/view/9154>
- List of airport codes Excel PDF. CopyLists.com. (2022, May 8). Retrieved March 5, 2023, from <https://copylists.com/geography/list-airport-codes/>
- Ngo, T. (2020, December 18). Airfares in New Zealand. Mendeley Data. Retrieved March 5, 2023, from <https://data.mendeley.com/datasets/w8wk4h9hxc>
- Wong, D. (2022, October 18). Flight prices. Kaggle. Retrieved March 5, 2023, from <https://www.kaggle.com/datasets/dilwong/flightprices>