Muhammad Shahmeer (19I-0622)

Aircraft Route Optimization with Genetic Algorithm

Project Report

Contents

[**1. Data Loading and Cleaning:** 2](#_Toc151588513)

[**2**. **Data Cleaning Check:** 2](#_Toc151588514)

[**3**. **Genetic Algorithm Functions:** 2](#_Toc151588515)

[ **initialize\_population** 2](#_Toc151588516)

[ **haversine:** 2](#_Toc151588517)

[ **get\_wind\_speed:** 2](#_Toc151588518)

[ **calculate\_fuel\_consumption:** 2](#_Toc151588519)

[ **calculate\_fitness:** 2](#_Toc151588520)

[ **tournament\_selection:** 2](#_Toc151588521)

[ **crossover** 2](#_Toc151588522)

[ **mutate** 2](#_Toc151588523)

[ **genetic\_algorithm** 3](#_Toc151588524)

[**4. Example Usage:** 3](#_Toc151588525)

[**5. Code Structure and Style:** 3](#_Toc151588526)

# **1. Data Loading and Cleaning:**

The code begins by importing necessary libraries and loading datasets (**aircraftDataset.csv**, **airportDataset.csv**, and **weatherDataset.csv**) using Pandas. Null values are then removed from each dataset, ensuring data cleanliness. Sample data for each dataset is displayed.

# **2**. **Data Cleaning Check:**

The script proceeds to check for missing values in each dataset, printing the count of null values in the Aircraft, Airport, and Weather datasets.

# **3**. **Genetic Algorithm Functions:**

The code defines several functions for a Genetic Algorithm (GA) that aims to find the best route for an aircraft journey considering factors like wind speed and fuel consumption.

* **initialize\_population**: Generates a random initial population of routes based on available aircraft ICAO codes.
* **haversine:** Calculates the great-circle distance between two geographical coordinates using the Haversine formula.
* **get\_wind\_speed:** Retrieves the average wind speed for a given route and date from the weather dataset.
* **calculate\_fuel\_consumption:** Estimates fuel consumption for a given route based on aircraft data, cruise speed, and wind speed.
* **calculate\_fitness:** Computes the fitness of a route, considering fuel consumption and wind speed. The fitness is inversely proportional to fuel consumption and directly proportional to wind speed.
* **tournament\_selection:** Implements tournament selection to choose individuals for crossover based on their fitness.
* **crossover:** Performs single-point crossover between two parents to produce two children.
* **mutate:** Introduces random mutations in an individual's genes (routes) with a given mutation rate.
* **genetic\_algorithm:** Combines the above functions to execute the Genetic Algorithm. It initializes a population, evaluates fitness, performs selection, crossover, and mutation iteratively over several generations.

# **4. Example Usage:**

The code concludes with an example usage of the Genetic Algorithm, prompting the user to input the source airport, destination airport, and date of the flight. This interaction with the user enhances the code's practicality and user-friendliness. The result is the best route calculated by the algorithm, providing a tangible output for the user.

# **5. Code Structure and Style:**

The code exhibits a well-organized structure with functions encapsulating specific functionalities. This modular approach enhances readability, maintainability, and reusability. Variable names are descriptive, contributing to code readability. Comments are interspersed throughout the code, explaining the purpose of each section, which is commendable for code maintainability.