**Future Challenge**

**Helping Balloons Navigate the Weather**

Anything for help: **bigdatacontest@alibabacloud.com**

1. Background

The scenario is set in the future—in the year of 2050, a world where the invention of “anti-gravity engines” has led to the creation of unmanned balloons as the preferred logistics solution. However, because of the UK’s complex meteorological conditions, the balloons are occasionally delayed, damaged or even destroyed by extreme weather conditions causing disruption and loss. The contestants will be challenged to create algorithms that can plan flight routes for these balloons to navigate the endless variation and changeable nature across the UK to optimize their delivery schedules and costs.

2. Objective

The objective of the contest is to develop an efficient navigation algorithm for cargo balloons while avoiding volatile weathers.  In the future not so far away, propelled cargo balloons set out daily from Hyde Park London to 10 destination cities in Britain.  If the wind is too strong, cargo balloons will be destroyed in midair.  As such, the flight paths of these balloons are planned before each flight, based on the weather forecast by the Met Office. However, the weather forecast is usually 90~95% accurate.  The Met Office runs 10 different forecast models every day and these each results in slightly different forecasts each more or less correct.  Given the daily weather forecasts, the winning algorithm will chart the shortest while safe paths for cargo balloons.

To simplify the challenge, we divided the coverage area into blocks according to a minimum resolution of weather forecasts. Each block is expressed by an x-axis coordinate and a y-axis coordinate (x,y), and we assume a constant air speed for cargo balloons under all weather conditions. The travel time across each block is fixed by two minutes and the balloons can move up, down, left, and right from the current block or stay at the current block, but cannot move diagonally.

At 3:00 every morning, 10 propelled cargo balloons set out from Hyde Park London to 10 destination cities in Britain. The travel duration is restricted to 18 hours [03:00-21:00]. Contestants are expected to forecast the weather condition of each (x,y) block so as to create the flight paths. For detailed data information, please refer to the next section. Only one condition is considered to cause flight crashes: wind velocity. When the wind velocity is ≥15, the balloon will crash.

3 Data

Four sorts of data files will be provided to the contestants: city data, weather forecast data, in-situ measurement weather data, and testing data. Some parts of the data sets have been anonymized. We provide 5 days' worth of weather forecast data and in-situ measurement weather data, a period equivalent to the testing data.

The city data is provided in “CitaData.csv” file, including the city IDs and x-axis and y-axis (x,y) coordinates of blocks. London is the origin city numbered as 0, while the rest are all destination cities numbered from 1 to 10. The format is shown in Figure 1.

Figure 1: City Data

|  |  |  |
| --- | --- | --- |
| City ID | x-axis | y-axis |
| 0 | 142 | 328 |
| … | … | … |

The weather forecast data and in-situ measurement weather data produced by the Met Office have been anonymized. The forecast data file is named as “ForecastDataforTraining.csv” which contains 6 columns as presented in Figure 2; and the in-situ measurement weather data file “In-situMeasurementforTraining.csv” contains 5 columns presented in Figure 3.

Figure 2: Weather Forecast Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x-axis | y-axis | Date ID | Forecast time (hour) | Model ID | Wind velocity |
| 22 | 201 | 2 | 14 | 1 | 4.91 |
| 45 | 32 | 1 | 21 | 2 | 1.28 |
| … | … | … | … | … | … |

Figure 3: In-situ Measurement Weather Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x-axis | y-axis | Date ID | Forecast time(hour) | Wind velocity |
| 120 | 25 | 3 | 14 | 6.35 |
| 82 | 19 | 5 | 21 | 3.56 |
| … | … | … | … | … |

The online testing data is named as “ForecastDataforTesting.csv”, the format of which is the same as the weather forecast data presented in Figure 2.

4. Player Submission

The player will submit one summary file in csv format of flight paths based on the 5-day testing data.  The summary file should contain five data columns: destination city ID, date ID, time (hh:mm), x-axis coordinate, and y-axis coordinate. The flight path file should contain the detailed flight path for every two minutes during the flight.

The submission file should look like Figure 4. Please ensure to exclude the header when you submit the file.

Figure 4: Submission sample

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Destination City ID | Date ID | time | x-axis | y-axis |
| 3 | 6 | 03:50 | 2 | 5 |
| 8 | 9 | 08:28 | 4 | 6 |
| 10 | 10 | 15:32 | 8 | 3 |

5. Objective Function

We will assess the balloon routes described in the contestants' submission against the actual hourly weather conditions during the day.  If on any day a balloon flies into calamitous weather (meaning it is destroyed), that contestant will receive a 24-hour delay penalty.

The final score on the leaderboard will be the total amount of travel time in minutes for all successful balloon flights, plus the total amount of penalties. The lowest score wins the leaderboard.

The objective function = 24\*60\*number of crashes + total travel minutes of successfully arrived balloons

Notes:

A. Contestant submissions should make use of only the datasets included in this contest.

B. Employees of Alibaba Cloud and the Met Office are not allowed to compete in this contest.

**Balloons Navigate**

问题描述：

MAP：地图为**548 x 421**，单位为1

无人机连续飞行每**2分钟**飞跃一个单元格，期间对天气进行预测

目标函数数值最小胜出

**Min{目标函数值}= 24\*60\*飞行器坠毁数 + 顺利到达的飞行器总飞行时长（分钟）**

假设如下：

1. 天气预报所覆盖的最小范围，对覆盖区域进行了区域块的划分，每一个区域块都可以用（x,y）唯一表示，x表示X轴方向的坐标值，y表示Y轴方向的坐标值。

2. 同时我们假设无人运输飞行器在所有天气条件下的飞行速度均保持不变，

3. 天气预报数据的间隔时间为1个小时，但无人机**2分钟**飞越一个区域块，一个小时的天气保持不变。

4. 风速。当风速值≥15时，无人机坠毁。

5. 每天3点钟10架推进式无人运输飞行器将从伦敦海德公园飞往英国其他10个目的地城市，限定最大飞行时长为18个小时 [03:00-21:00]。

思路：

一个小时内可以走