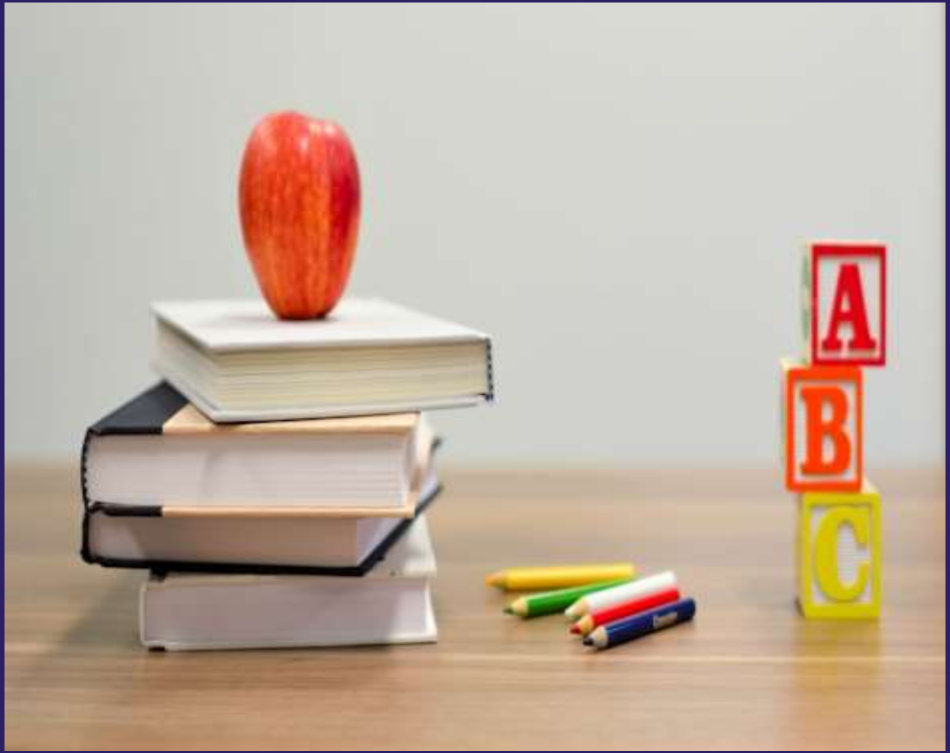
Visualizations



Created by Mikheil Kvizhinadze

Leveraging the Google Maps Directions API

Route Planning: We utilized the Google Maps Directions API to obtain an optimized route between a specified origin and destination.

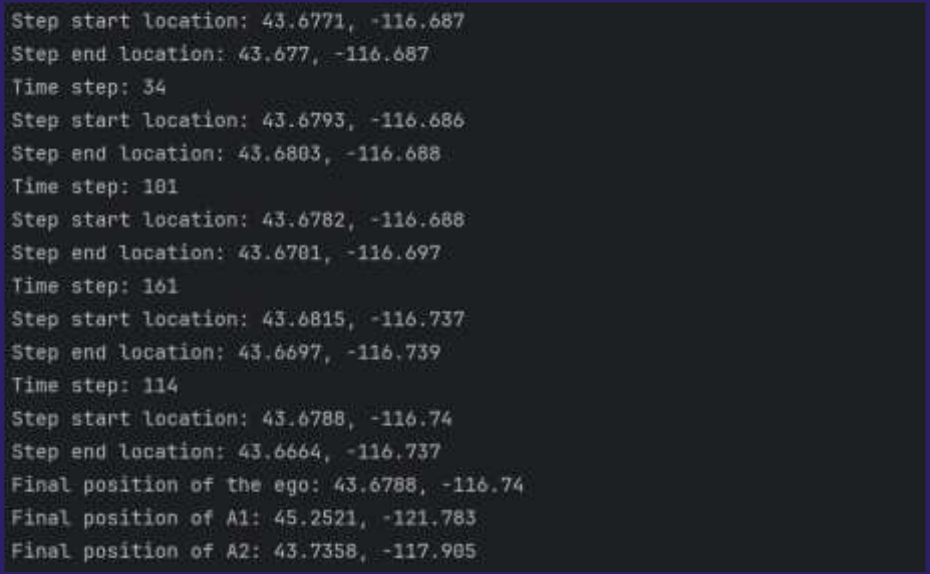
Simulation Foundation: We used the API's route information as a baseline to build our movement simulation, which factored in the ego's specific speed.

Console output and explanation

Step Division: The route provided by the Directions API consists of steps with durations exceeding 3 seconds. To accurately check the special condition (function value being 100% higher), we need finer-grained movement simulation.

1-Second Substeps: We divide each route step into 1-second substeps. This enables us to:

> Track the ego's position with higher precision and
> evaluate the special condition (f_c, f_1, f_2) at a frequency consistent with the 3-second requirement.



```
Step start location: 43.6771, -116.687
Step end location: 43.677, -116.687
Time step: 34
Step start location: 43.6793, -116.686
Step end location: 43.6803, -116.688
Time step: 101
Step start location: 43.6782, -116.688
Step end location: 43.6781, -116.697
Time step: 161
Step start location: 43.6815, -116.737
Step end location: 43.6697, -116.739
Time step: 114
Step start location: 43.6788, -116.74
Step end location: 43.6664, -116.737
Final position of the ego: 43.6788, -116.74
Final position of A1: 45.2521, -121.783
Final position of A2: 43.7358, -117.905
```

Part of console output

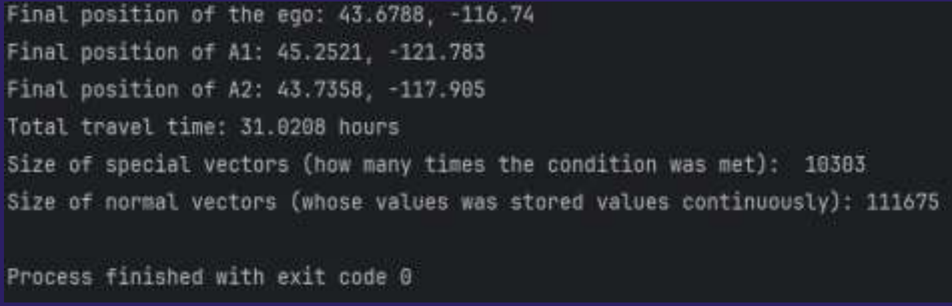
Improved Accuracy: Subdividing steps enhances the simulation's realism by modeling the ego's continuous movement rather than assuming instantaneous jumps at the start and end of each step.

Data Granularity: The 1-second substeps provide a consistent time interval between data points (ego position, function values).

Console output and explanation

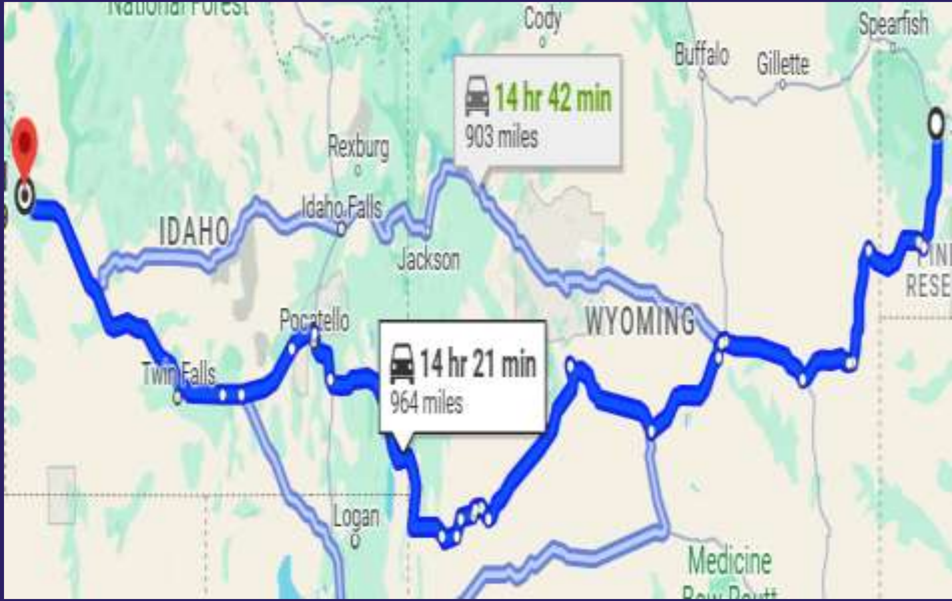
Values of ego location, function values, and distances to, destination, center of C, center of A1 and center of A2 are stored every second, so we have 111675 values and total travel time is 111675 / 3600 = 30.02 hours.

Moreover, total travel time is consistent with predicted value, 964 * 1.6 / 50 = 30.85 hours. We will get this value if we follow one of the routes suggested by Google Maps and if our velocity will be 50km/hour (constant)



```
Final position of the ego: 43.6788, -116.74
Final position of A1: 45.2521, -121.783
Final position of A2: 43.7358, -117.905
Total travel time: 31.0208 hours
Size of special vectors (how many times the condition was met):  10303
Size of normal vectors (whose values was stored values continuously): 111675

Process finished with exit code 0
```



Choice of tools for visualizations

Python:

Data-centric: Offers robust libraries (like Pandas) for efficient data manipulation and analysis.

Visualization Libraries: Provides mature plotting libraries like Matplotlib, enabling the creation of clear and informative visualizations.

Jupyter Notebook:

Interactive Development: Streamlines visualization development with the ability to iteratively code, plot, and refine visualizations within the same environment.

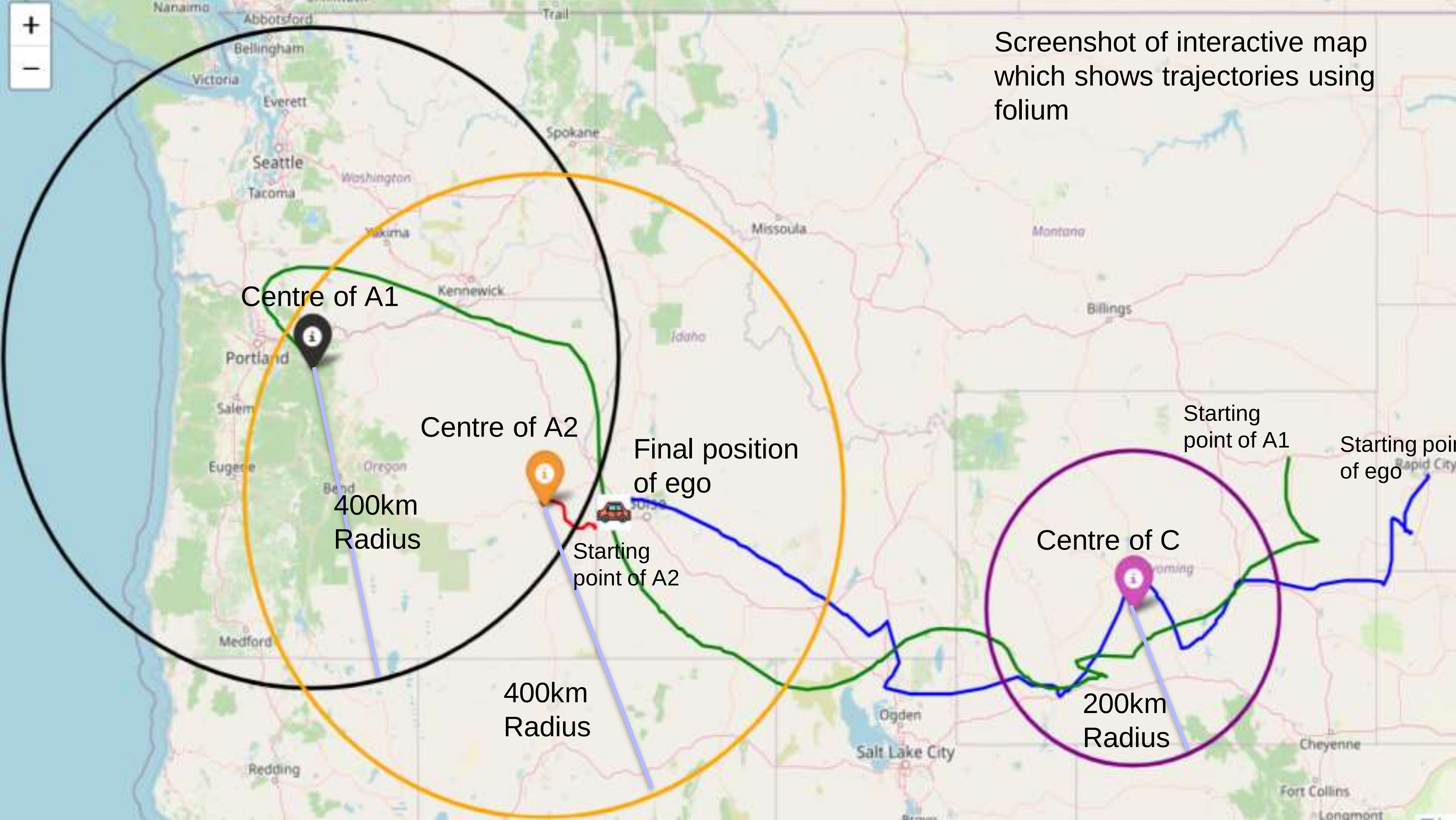
Enhanced Storytelling: Eases communication of analysis and results by combining code, visualizations, and explanatory text.

Folium:

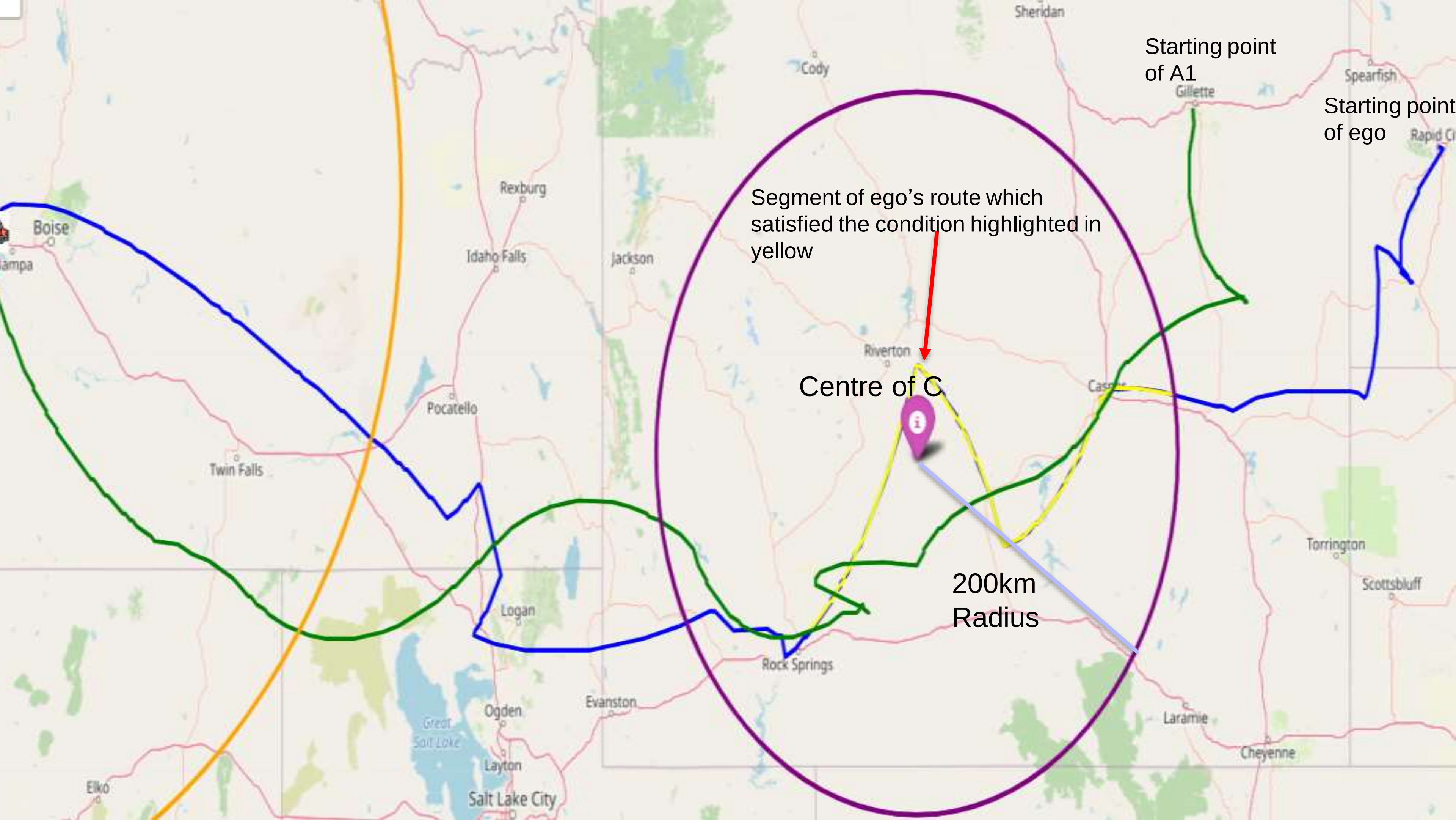
Simplifies the creation of interactive maps directly within Python, seamlessly integrating route and location data visualization.

Provides flexibility in customizing map elements to effectively communicate findings and highlight spatial relationships.

Next slide shows screenshot of routes on a map obtained using folium library



Screenshot of interactive map which shows trajectories using folium



Starting point of A1

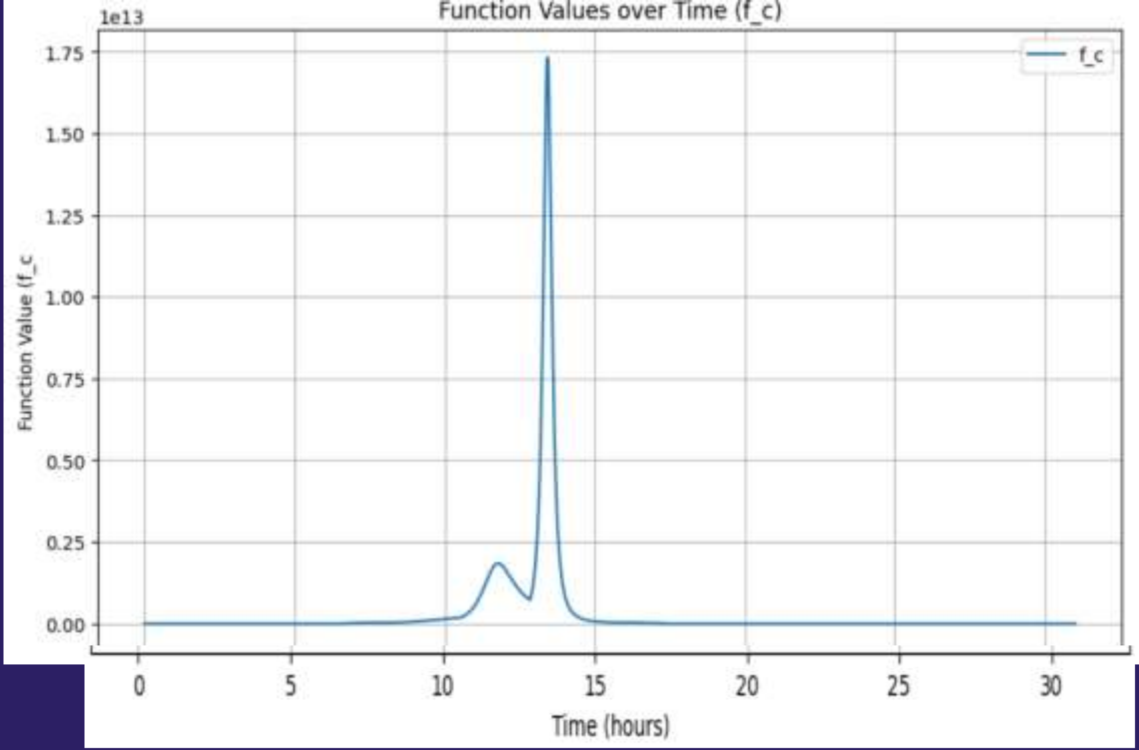
Starting point of ego

Segment of ego's route which satisfied the condition highlighted in yellow

Centre of C

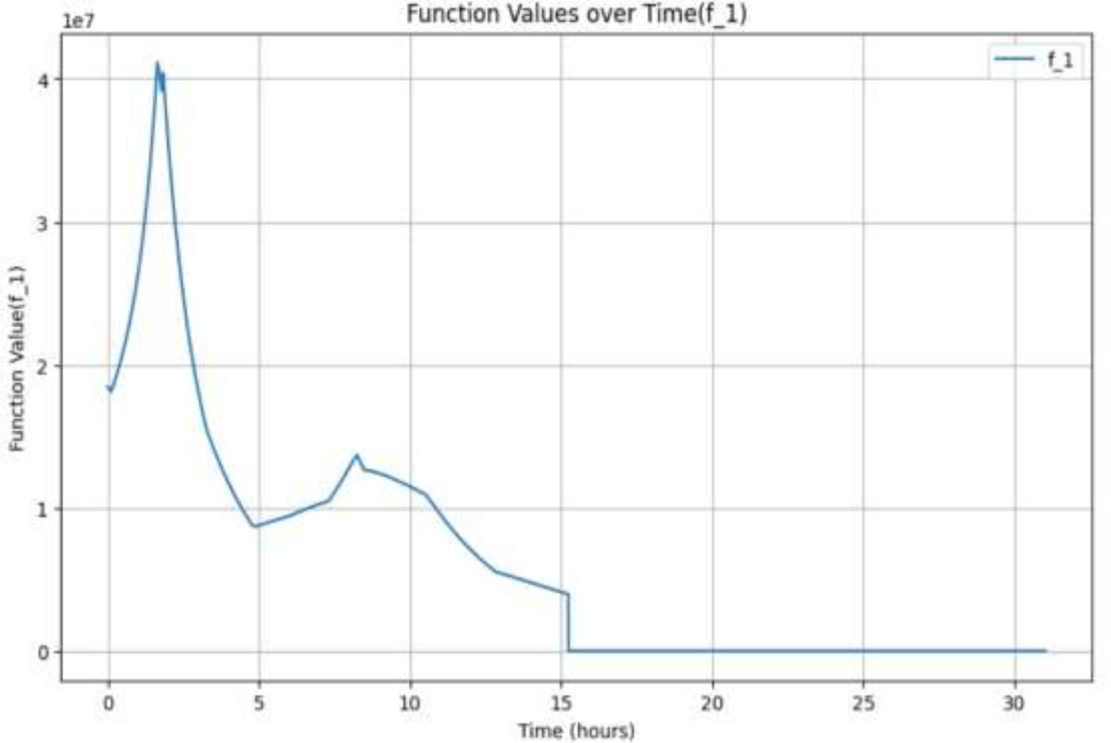
200km Radius

Function values over time(f_c)



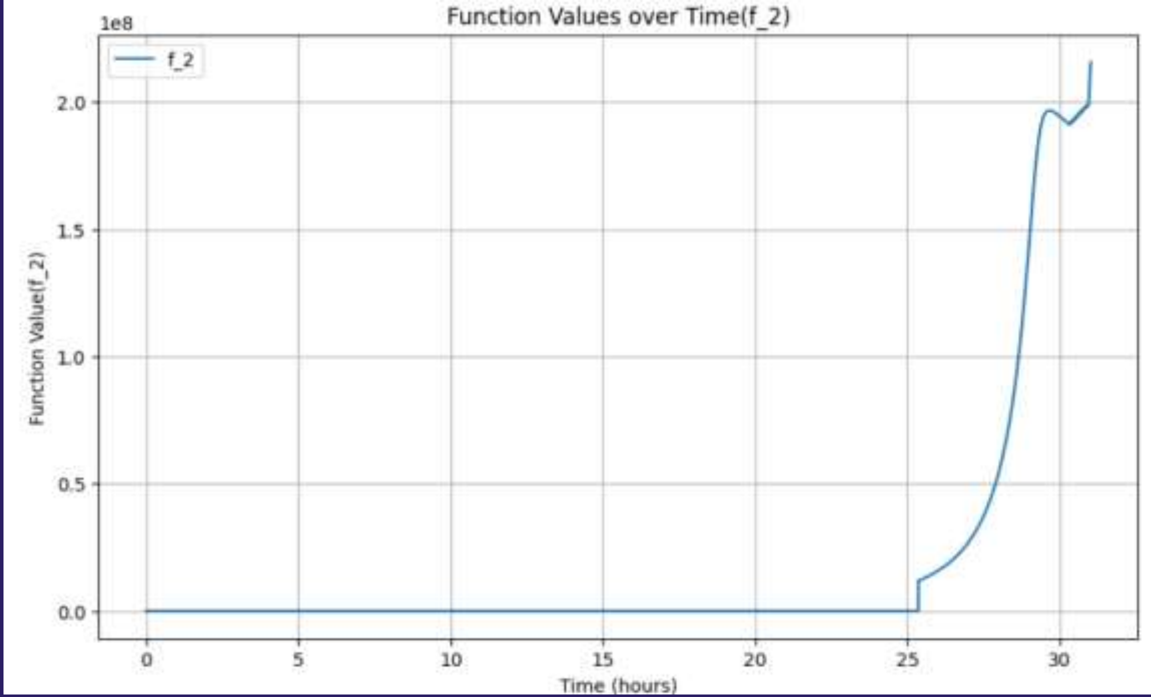
The circle denoted as 'C' remains static throughout the scenario. Consequently, non-zero values are observed only when the ego vehicle enters into the circle ‘C’. Outside of these instances, the recorded values revert to zero.

Function values over time(f_1)



Initially, the ego vehicle resides within the confines of Circle A1, resulting in a non-zero function output. However, as Circle A1 outpaces the ego vehicle, the function steadily declines over time until it reaches zero.

Function values over time(f_2)



In the beginning, the Circle A2 and the ego vehicle are considerably distant from each other. So, the function remains at zero for the majority of the duration. However, as Circle A2 moves much slowly, than the ego vehicle, so eventually, ego vehicle enters Circle A2 and function f2 becomes non-zero, as expected.

Mikheil Kvizhinadze