**Thesis Status (as on 3rd February, 2019)**

* **Tasks Completed :** 
  + Finding the shortest Route with the extra parameter added to the dijkstra’s Algorithm as an overhead (for example: road\_width, road\_construction, etc.)
  + Finding the time taken by the normal dijkstra algorithm, dijkstra-with-congestion algorithm and dijkstra-with-overhead algorithm.
  + Comparision of different programs used in SUMO and the times taken by the program to find the shortest route from origin to destination. Four algorithms are inherently used in SUMO – Dijkstras algorithm, A\* algorithm, CH algorithm and CHWrapper algorithm.
  + Added the parser in the former script that I have made for computing distance with the dijkstras algorithm taking congestion into consideration.
  + Simulation of all the Scenarios in SUMO-GUI application:-
    - Dijkstras Algorithm
    - Dijkstras Algorithm taking congestion into consideration.
    - Dijkstras Algorithm taking overhead into consideration.
* **Details :**
  + Added a new function for parsing the new argument in the taz-file and its usage in the main program
  + Calculation of the time taken by the shortestPath(arg1,arg2,arg3)

|  |  |  |
| --- | --- | --- |
| **S.no.** | **Algorithm** | **Time-Taken (in seconds)** |
| 1 | Dijkstra Algorithm | 0.00800013542175 |
| 2 | Dijkstra Algorithm with congestion | 0.04100012779240 |
| 3 | Dijkstra Algorithm with overhead | 0.06799983978270 |

* + Comparison of different inherent algorithms that are provided by the SUMO are as below :-

|  |  |  |
| --- | --- | --- |
| **S.no.** | **Algorithm** | **Time-Taken (in seconds)** |
| 1. | Dijkstras Algorithm | 0.00300 |
| 2. | CHWrapper Algorithm | 0.00200 |
| 3. | CH Algorithm | 0.00054 |
| 4. | A\* Algorithm | 0.00100 |

* + Now we can simulate all the three scenarios through the parser by just writing **python Dijkstra\_with\_parser.py –routing\_algorithm <“dijkstra”/”dijkstra-with-congestion”/”dijkstra-with-overhead”>**
  + Attaching all the source files and the main logic behind the simulation in the zip file. Please find it in the encloser.

Thankyou