MATSIM

INTERNSHIP REPORT

Submitted in partial fulfillment of the requirements for the award of the degree

Of

-BACHELOR OF TECHNOLOGY-

In

-COMPUTER SCIENCE AND ENGINEERING-

By

-MADHU PATEL-
-Patelmadhu06@gmail.com-

Guided by

-Amit Agarwal-
-Assistant Professor-
-Civil Engineering, IIT Roorkee-
-amit.agarwal@ce.iitr.ac.in-

- MAY2023 -

ACKNOWLEDGEMENT

I thank Prof. Amit Agarwal for being my Faculty Mentor and for helping me decide what my project was going to be. His talks have been a great help both technically and otherwise.
I also thank Shivam Arora who, as my Student Mentor, provided valuable and timely help and guided me throughout the project. He provided me with whatever resources I needed for my project and kept in constant contact.

Madhu Patel

CERTIFICATE

This is to certify that Madhu Patel of Indira Gandhi Delhi Technical University of Women of Technology, has completed the internship under the guidance of Prof Amit Agarwal and submitted the report titled MATSIM Development and submitted this report in partial fulfillment of the requirements of the Civil Department, Indian Institute of Technology Delhi.

Madhu Patel

LIST OF ABBREVIATIONS

Abbreviation	Description
MATSIM	Multi-Agent Transport Simulation
GIS	Geographic Information Systems
DTA	Dynamic Traffic Assignment
API	Application Programming Interface
OSM	OpenStreetMap
XML	Extensible Markup Language
CSV	Comma-Separated Values

TABLE OF FIGURES

CONTENTS

ABSTRACT

This report explores the use of MATSim, a multi-agent transport simulation software, for analyzing and improving transportation systems. It discusses the MATSim coordinate system basics, events, and demand generation. The report also covers creating different modes in MATSim and event handling. Additionally, the report details the process of creating test cases to compare the travel time of links and lanes and to analyze congestion by comparing total delay, waiting time, and average delay per link and average waiting per signal for different cases. Overall, this report provides a comprehensive guide to using MATSim for transportation system analysis and optimization.

Madhu Patel

CHAPTER 1: INTRODUCTION

MATSim (Multi-Agent Transport Simulation) is an open-source, modular, and scalable transport simulation platform that can be used to model and simulate the behavior of individuals in large-scale transportation systems. MATSim is designed to simulate the decision-making processes of agents, such as commuters, households, and businesses, in order to predict the impacts of policy changes and infrastructure improvements on travel behavior and traffic patterns.

1. Agent-based modeling: MATSim uses an agent-based modeling approach, which means that the system models individual agents and their decision-making processes. Each agent is represented as a software object that can make decisions about its travel behavior based on a set of rules and preferences.

2. Activity-based modeling: MATSim also employs an activity-based modeling approach, which means that the system models the entire travel chain for each agent. This includes the activities that agents engage in (such as work, shopping, and leisure), the locations where those activities occur, and the travel modes that agents use to get from one location to another.

3. Dynamic traffic assignment: MATSim incorporates a dynamic traffic assignment (DTA) module that simulates the flow of traffic on the transportation network in real-time. This allows the system to capture the impacts of congestion and other network-level effects on travel behavior.

4. Integration with other software: MATSim is designed to be modular and can be integrated with other software tools, such as geographic information systems (GIS) and travel demand models, to enhance its functionality and applicability.

5. Applications: MATSim has been used in a variety of applications, such as predicting the impacts of new transportation infrastructure, evaluating the effectiveness of congestion pricing policies, and designing new public transport services.

CHAPTER 2: MATSIM

2.1 Running MATSim

Building and running a basic scenario in MATSim involves several steps, including data preparation, network creation, population synthesis, scenario configuration, and simulation. Here is a brief overview of each step, along with an example of a simple MATSim scenario.

1. Data Preparation: The first step in building a MATSim scenario is to prepare the data required for the simulation. This includes data on the transport network, land use, travel demand, and other relevant factors. In the case of a simple scenario, we might start with a network file in a format that can be read by MATSim, such as OpenStreetMap data. We might also have data on travel demand in the form of origin-destination matrices or trip lists.

2. Network Creation: Once we have the network data, we need to create a MATSim network object that can be used in the simulation. This involves converting the network data into a format that MATSim can read, such as a MATSim XML file. The network object contains information on the network topology, including nodes, links, and lanes.

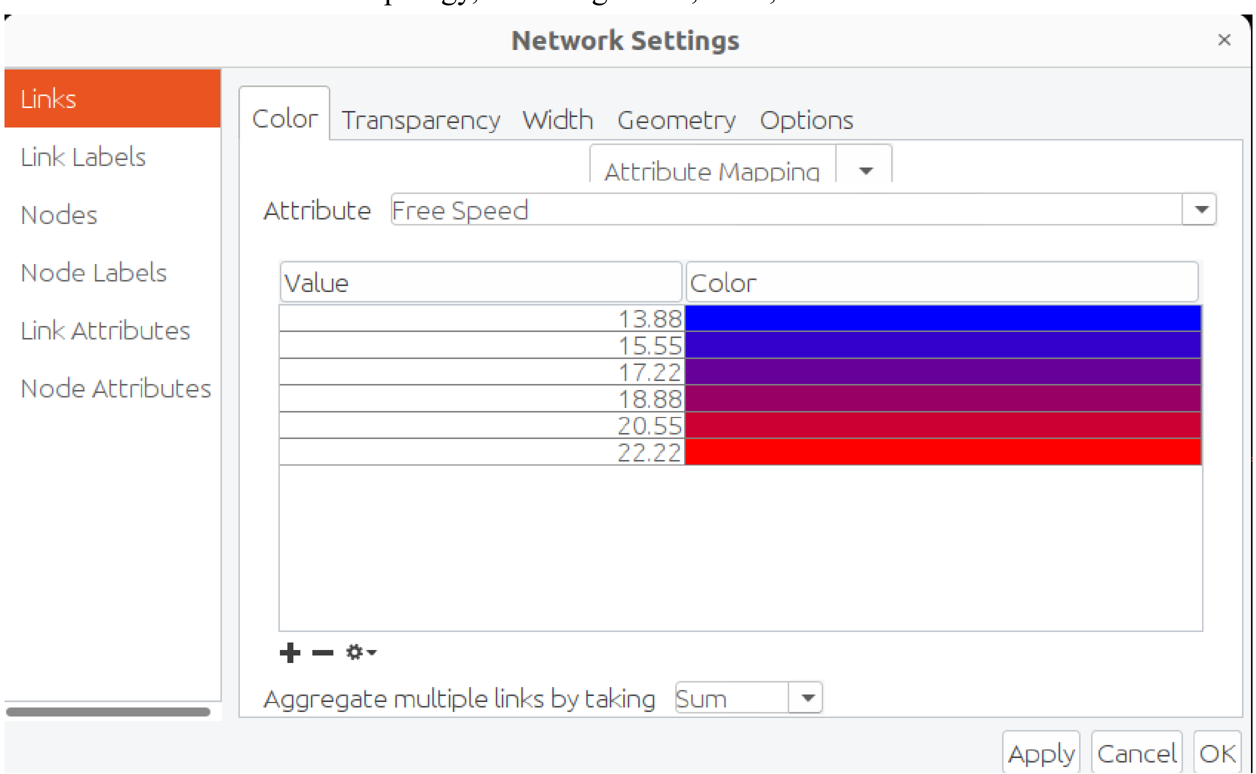


Figure 1: Networks settings in Via

3. Population Synthesis: The next step is to create a synthetic population of agents that will be used in the simulation. This involves generating a set of agents that represent the population of the study area. The population can be created using a variety of methods, such as random sampling or statistical modeling. For our simple scenario, we might create a population of 1000 agents with random attributes for age, income, and travel behavior.

4. Scenario Configuration: Once we have the network and population data, we need to configure the simulation scenario. This involves specifying the simulation parameters, such as the number of iterations, the start time, and the end time. We also need to specify the initial conditions for the simulation, such as the number of vehicles and the initial agent locations.

5. Simulation: Once the scenario is configured, we can run the simulation using MATSim software. The simulation involves iteratively simulating the behavior of the agents over a specified time period, based on their preferences and the network conditions. The output of the simulation includes data on the travel behavior of the agents, such as their travel times, mode choices, and route choices.

Figure 2: Situation of Network file of Delhi

Example Scenario: To illustrate these steps, let's consider a simple scenario of a part of Delhi City. And create the network file, population file, and Scenario Configuration.

To create the network object, we convert the network data into a MATSim XML file. We also create a synthetic population of 1000 agents using a statistical model that assigns each agent a random age, income, and travel behavior.

Network Simulation

Figure 3: Network file and simulation of two-way network

We then run the simulation using MATSim software. The simulation outputs data on the travel behavior of the agents, such as their travel times, mode choices, and route choices. We can analyze this data to gain insights into the impacts of different transport policies and infrastructure improvements on travel behavior and traffic patterns in the study area.

2.2 MATSim Coordinate System

When working with MATSim, it is important to specify the coordinate system of the data being used. This is necessary for some operations, like visualizing data in Google Earth or QGIS. One can specify their coordinate system in the config file using an EPSG code, which is a standardized number that identifies the coordinate system being used. The correct EPSG code can be found easily by searching for it on the website http://www.spatialreference.org. Alternatively, one can describe their coordinate system in the WKT format. This will enable MATSim to work with the coordinates and make necessary conversions.

```
< module name= " global " >
```

```
< param name= " coordinateSystem " value= " EPSG :32608 " / >
</ module >
```

2.3 MATSim Events

MATSIm is a powerful simulation tool for multi-agent transport systems, and it works by simulating the behavior of individuals as they travel through the transportation network. One key component of MATSim is the use of events to manage the various actions that occur within the simulation.

Events are essentially signals that inform the simulation when certain actions occur, such as the arrival of an individual at a particular location or the start of a new day. MATSim provides several built-in event types that can be used to manage these actions, but users can also create custom events to manage specific needs within their simulation.

One of the most common event types in MATSim is the ActivityEndEvent. This event is triggered when an individual completes an activity, such as arriving at work or finishing a shopping trip. The following code example shows how to create an ActivityEndEvent in MATSim:

```
public class MyEventHandler implements EventHandler {

  @Override
  public void handleEvent(Event event) {
    if (event.getEventType().equals(ActivityEndEvent.EVENT_TYPE)) {
      ActivityEndEvent activityEndEvent = (ActivityEndEvent) event;
      // Do something when an activity ends
    }
  }
}
```

In this example, the MyEventHandler class implements the EventHandler interface, which is used to manage all events within MATSim. The handleEvent method is called whenever an event occurs, and it checks whether the event is an ActivityEndEvent. If it is, the event is cast to an ActivityEndEvent object so that its properties can be accessed.

Another important event type in MATSim is the LinkEnterEvent. This event is triggered when an individual enters a link in the transportation network. The following code example shows how to create a LinkEnterEvent in MATSim:

```
public class MyEventHandler implements EventHandler {

  @Override
  public void handleEvent(Event event) {
    if (event.getEventType().equals(LinkEnterEvent.EVENT_TYPE)) {
      LinkEnterEvent linkEnterEvent = (LinkEnterEvent) event;
      // Do something when an individual enters a link
    }
  }
}
```

In this example, the MyEventHandler class is again used to manage events within MATSim. The handleEvent method checks whether the event is a LinkEnterEvent, and if it is, the event is cast to a LinkEnterEvent object so that its properties can be accessed.

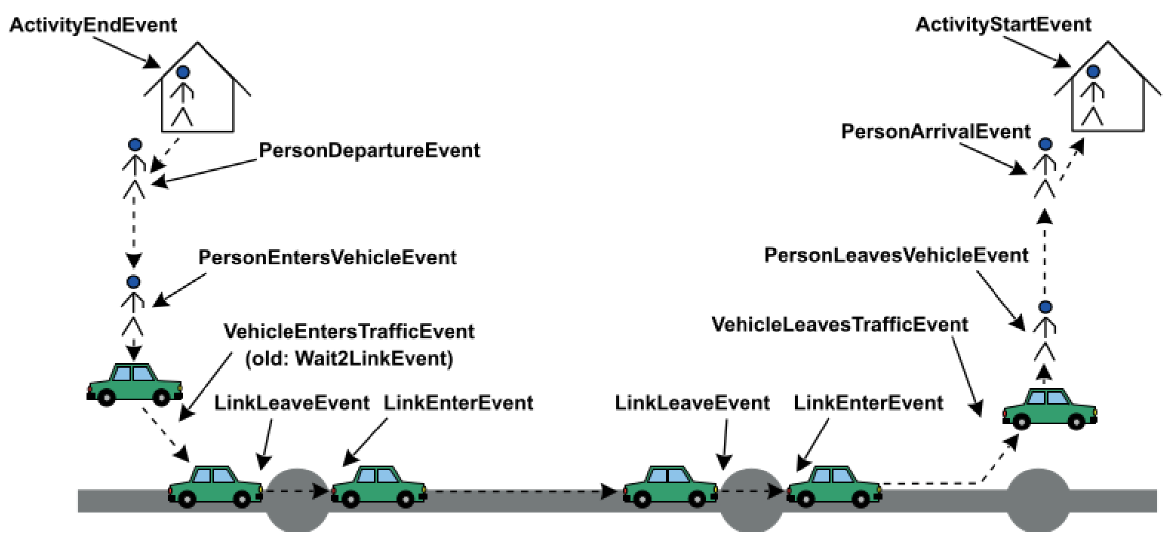


Figure 4: Mobsim events

In MATSim, events are handled using the Event API, which provides a set of classes and methods for defining and processing events. The Event API is based on the observer pattern, where objects can register themselves as observers of certain events and receive notifications when those events occur.

To use the Event API in MATSim, the first step is to define the events that you want to handle. This involves creating a new class that extends the Event class and implementing the handle method. The handle method contains the code that will be executed when the event is triggered.

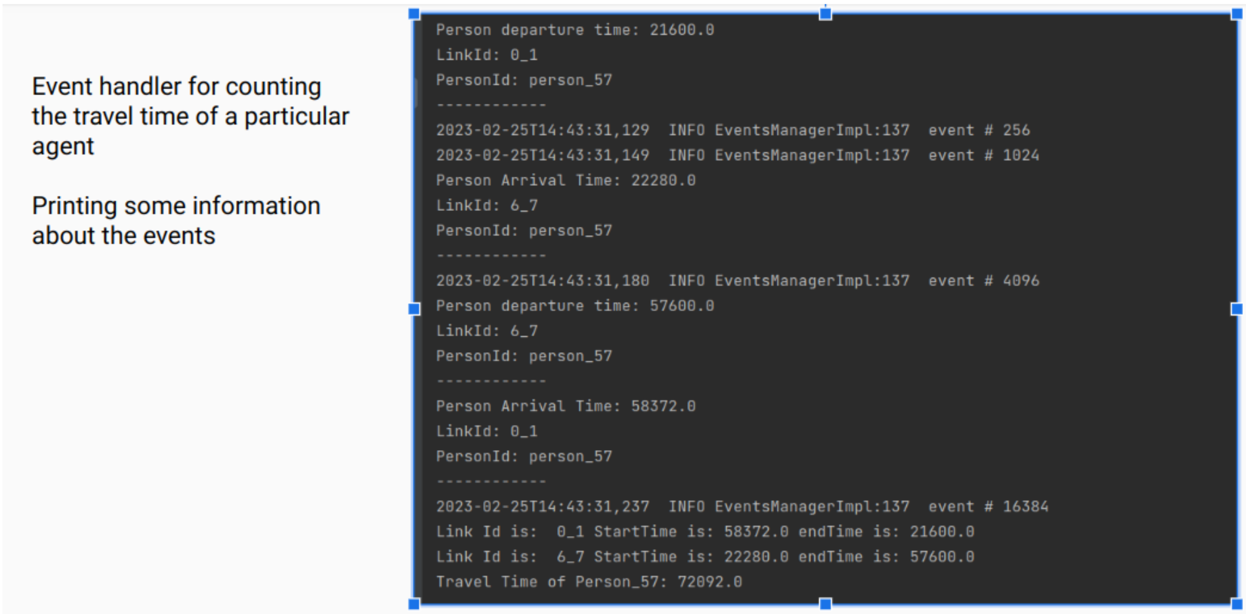


Figure 4: output of event handling

2.4 Congestion

MATSim allows for the simulation of congestion in a realistic and customizable way. Congestion is typically modeled by introducing travel time penalties based on the degree of congestion on a particular route. The level of congestion can be measured in a variety of ways, such as vehicle counts, speeds, or travel times.

One approach to modeling congestion in MATSim is to use a traffic flow model that simulates the flow of vehicles through a network of roads. This model takes into account the capacity of each road segment and the number of vehicles that are attempting to use it. Based on these inputs, it can calculate the flow of traffic and the resulting travel times for each route.

One of the advantages of using MATSim to model congestion is that it allows for the evaluation of various policies and interventions aimed at reducing congestion. For example, policymakers can simulate the impact of introducing tolls or congestion charges, implementing traffic signal optimization, or investing in public transportation.

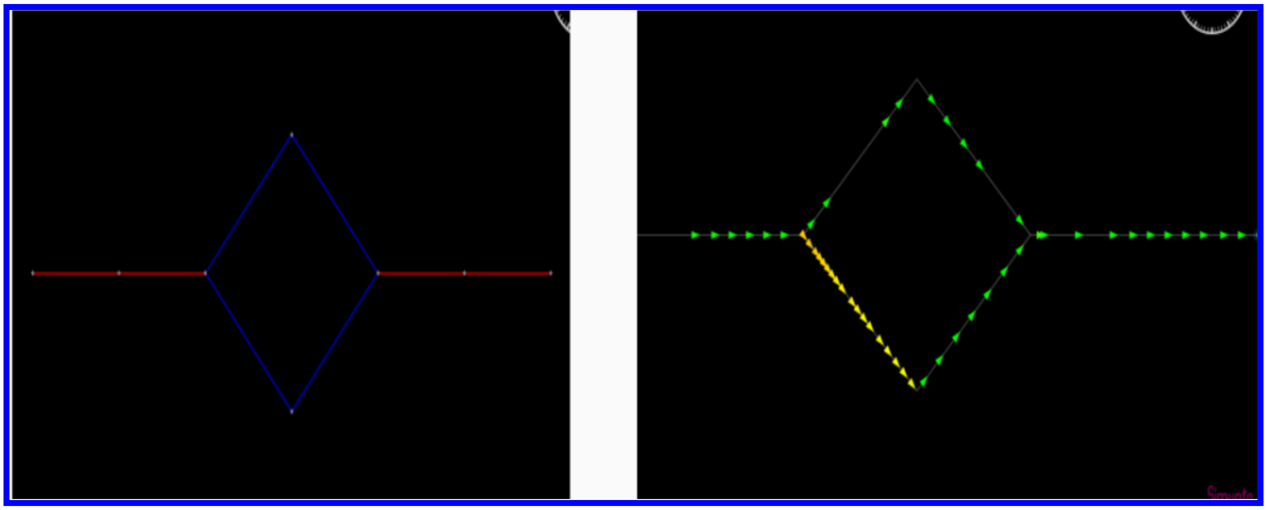


Figure 5: Network creation and simulation

https://drive.google.com/file/d/1WZryVI5vRlEwNuMH0qT75nPo9xiUoOpd/view?resourcekey

CHAPTER 3: WORK CONTRIBUTIONS

3.1 Generating Demand

It involves various steps like

1. Data preparation: The first step in generating demand is to collect and prepare the necessary data. This includes data on the population, land use, transportation infrastructure, and travel behavior. The data should be in a format that can be used by MATSim, such as XML or CSV.

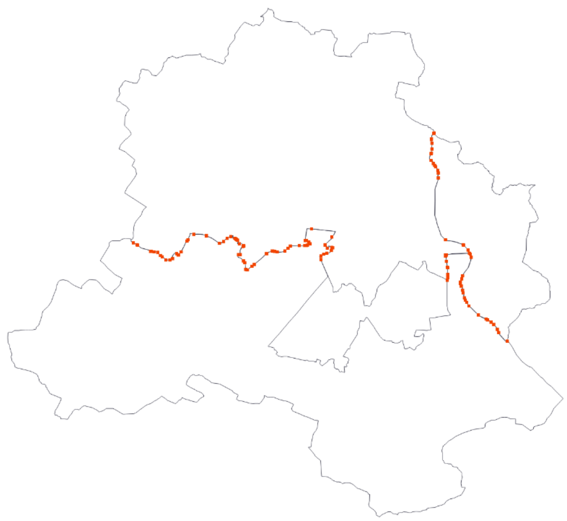


Figure 6: Shapefile of Delhi

2. Population synthesis: The next step is to create a synthetic population that represents the real population in the study area. This involves using statistical methods to generate a population that matches the characteristics of the real population in terms of demographics, socioeconomic status, and travel behavior.

3. Activity generation: Once the population has been synthesized, the next step is to generate activity schedules for each person in the population. This involves assigning activities to each person, such as work, school, shopping, and leisure, and specifying the time and location of each activity.

4. Trip generation: Based on the activity schedules, the next step is to generate trips for each person in the population. This involves determining the origin and destination of each trip, as well as the mode of transportation that will be used.

5. Route assignment: The final step in generating demand is to assign a route to each trip. This involves using a route assignment algorithm to find the most efficient route between the origin and destination, taking into account factors such as travel time, distance, and congestion.

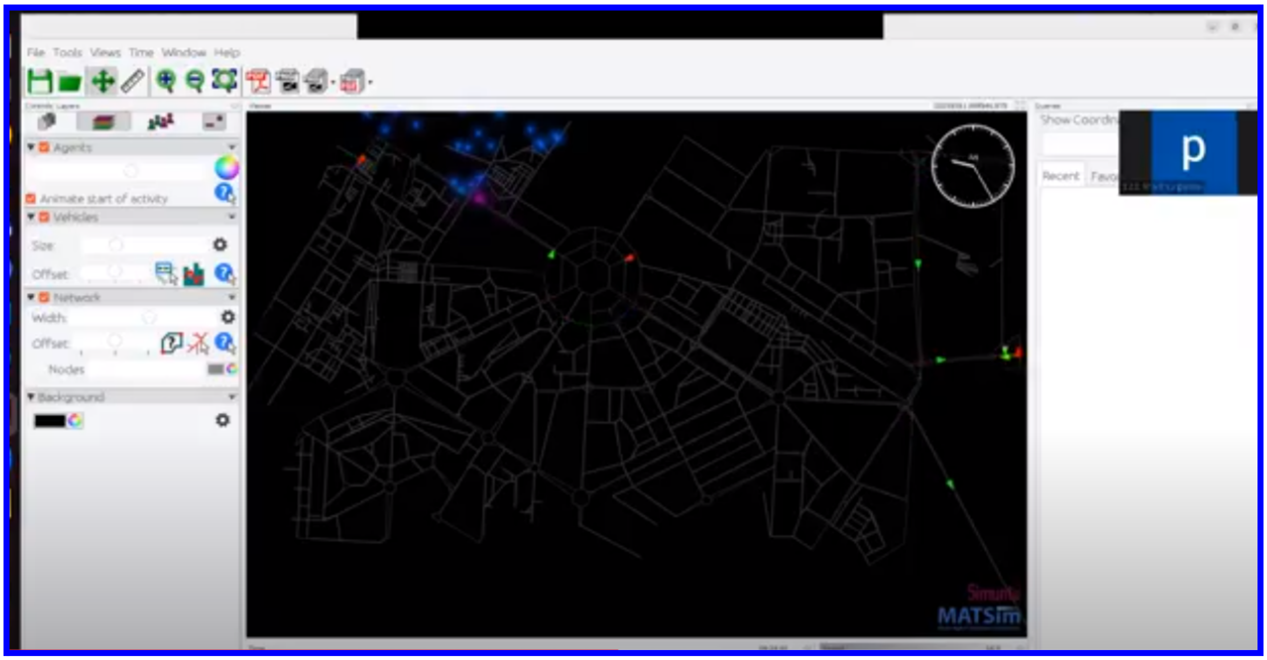


Figure 7: Simulation: generating demand

https://drive.google.com/file/d/1ZNu2zZnv3AZJb--526hgB4JmpMqi2cMe/view?resourcekey

https://drive.google.com/file/d/1ZLbd8sZRSmgiqbI0rgHbfijAavtuZDt0/view?resourcekey

MATSim provides tools and modules to perform each of these steps. For example, the Population module can be used to synthesize the population, the ActivityGen module can be used to generate activity schedules, and the TripGen module can be used to generate trips. In addition, MATSim provides several route assignment algorithms, such as Dijkstra, A*, and Contraction Hierarchies, that can be used to assign routes.

3.2 Creating different Modes in MATSIM

Define the mode: First, you need to define the mode in the modes.xml file. This file contains information about the different modes used in the simulation, such as the name of the mode, the travel costs, and the vehicle types.

Define the vehicle type: In the vehicleTypes.xml file, you can define the different vehicle types for each mode. This includes information about the size, weight, and capacity of the vehicle.

Define the network: The network file contains information about the roads, intersections, and other infrastructure used in the simulation. You need to specify which modes are allowed on each road segment and the speed limit for each mode.

Define the travel costs: The modes.xml file also contains information about the travel costs for each mode. This includes the cost per unit distance, such as per kilometer or per minute.

Assign modes to agents: In the plans.xml file, you can assign the different modes to the agents in the simulation. This includes specifying the mode for each leg of the agent's trip, such as walking, cycling, or driving.

```
{
        String mode = "bike";
        PlanCalcScoreConfigGroup.ModeParams params= new
PlanCalcScoreConfigGroup.ModeParams(mode);
        config.planCalcScore().addModeParams( params );
 }

{
         String mode = "bicycles";
         PlanCalcScoreConfigGroup.ModeParams params= new
PlanCalcScoreConfigGroup.ModeParams(mode);
         config.planCalcScore().addModeParams( params );
}
```

```
VehicleType rides = VehicleUtils.getFactory().createVehicleType(Id.create("ride",
VehicleType.class));
      rides.setMaximumVelocity(15.0);
      rides.setPcuEquivalents(0.05);
      rides.setNetworkMode("ride");
      scenario.getVehicles().addVehicleType(rides);
```

Figure 8: Simulation: creating different modes in MATSIM

Code Contributions in: https://github.com/Madhupatel08/MATSIM-Internship

3.3 Creating Test Cases for Comparing Travel Time of Links and Lanes

1. Generate the demand: Generate the demand for the selected scenarios using the MATSim demand generation tools. This involves specifying the number of agents, their start and end points, and their preferred mode of transport.
2. Run the simulations: Run the simulations for the selected scenarios using the MATSim simulation tools. This involves running the simulations for the selected time period and collecting the data on travel times for the selected links and lanes.
3. Analyze the data: Analyze the data collected from the simulations and compare the travel times of the links and lanes using the metrics defined in step 2. This can be done using the MATSim analysis tools.
4. Validate the results: Validate the results by comparing them with the actual travel times observed in the selected area. This can be done by collecting data on travel times using GPS or other tracking devices.

Code Contributions in: https://github.com/teg-iitr/matsim-iitr/tree/madhu

3.4 Creating Test Cases and comparing results for Totaldelay, waiting, and average delay per link and average waiting per signal for different cases

Totaldelay refers to the sum of all the delays experienced by vehicles traveling on a specific link or set of links during a given period. This metric helps in identifying bottlenecks and congested areas of the network.

Waiting time refers to the amount of time that a vehicle spends waiting at a signal or intersection. This metric is important as it directly affects the travel time of a vehicle and the overall flow of traffic.

Average delay per link is the average delay experienced by vehicles on a specific link over a given period. This metric helps in identifying links with the highest delay and can aid in the planning of measures to reduce congestion and improve traffic flow.

Average waiting per signal refers to the average waiting time for vehicles at a signal or intersection. This metric is useful in identifying signals that cause the most delay and can be used to optimize signal timings to reduce waiting time and improve traffic flow.

Code Contributions in: https://github.com/teg-iitr/matsim-iitr/tree/compareResults

Adaptive strategies in MATSIM are more effective when the lane capacity is lower than the link capacity. This is because in such cases, the adaptive approach can more efficiently allocate resources to reduce congestion. The effect of link length on congestion is still not very clear and requires more testing to determine its impact. However, the storage factor and flow capacity factor are significant factors that play a role in congestion. Therefore, adaptive strategies are better suited to handle congestion in such situations, as they can adjust the storage and flow capacity factors as needed. Finally, increasing demand inevitably leads to increased congestion. In such situations, adaptive strategies become more effective in managing congestion compared to fixed strategies.

3.5 Comparing Fixed Time Mixed-Signal and Adaptive Mixed-Signal

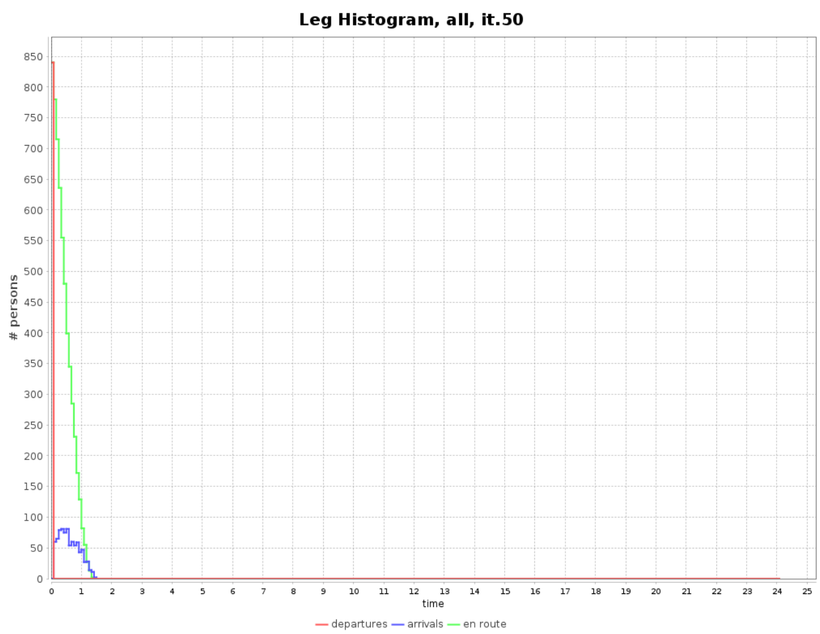
we conducted an analysis to evaluate the performance of two different traffic signal control methods.

Code Contributions in: https://github.com/teg-iitr/matsim-iitr/tree/compareResults

To assess the impact of various factors on the traffic flow, we manipulated different parameters, such as AGENTS_PER_LEFT_APPROACH, AGENTS_PER_TOP_APPROACH, AGENTS_PER_RIGHT_APPROACH, AGENTS_PER_BOTTOM_APPROACH, ITERATION, STORAGE_CAPFACTOR, FLOW_CAPFACTOR, and CYCLE Length. By varying these parameters, we aimed to observe the corresponding changes in the average delay time experienced by vehicles on the road links.

Furthermore, during the analysis, we ensured that the agents (representing vehicles) were uniformly distributed over the course of one hour. This approach was adopted to facilitate smoother traffic movements and minimize congestion on the road network. By evenly distributing the agents, we aimed to create a more realistic traffic simulation and capture the dynamics of traffic flow under different control methods accurately.

Results:- The leg diagram for one of the iterations is shown here.



- The Fixed-Time Mixed Signals Perform better as compared to Adaptive Mixed Signals.

CHAPTER 4: RESULTS & CONCLUSION

1) Working on MATSIM can provide valuable insights into the behavior of complex transportation systems and aid in policy and planning decisions. The ability to generate and analyze large amounts of data using MATSIM can help researchers and policymakers identify potential solutions to transportation problems, such as congestion and emissions, and evaluate their effectiveness.
2) Adaptive is better when lane capacity is lower than link capacity
3) Link Length does not have an effect
) The Fixed-Time Mixed Signals Perform better as compared to Adaptive Mixed Signals.

CHAPTER 5: FUTURE SCOPE

1. More Testing can be done to check the effect of Link Length and Lane Length on traffic congestion and adaptive traffic management systems.
2. Real-time simulation and control: Real-time simulation and control could be added to MATSim to allow for more dynamic and responsive transportation management. This would allow for more effective management of traffic flow, congestion, and incidents.
3. Integration with autonomous vehicle technology: As autonomous vehicle technology continues to advance, MATSim could be used to simulate the impact of autonomous vehicles on traffic flow and infrastructure demand. This could include scenarios such as fleet management, traffic coordination, and modeling interactions between autonomous and non-autonomous vehicles.
4. Integration with multimodal transportation systems: MATSim could be extended to model multimodal transportation systems that include public transportation, shared mobility services, and non-motorized transportation modes such as biking and walking. This would allow for a more comprehensive analysis of transportation planning and management scenarios that involve multiple transportation modes.

BIBLIOGRAPHY

1. Horni, A., Nagel, K., & Axhausen, K. W. (2016). The Multi-Agent Transport Simulation MATSim. In Handbook of Transport Modelling (pp. 1-25). Springer, Cham. DOI: 10.1007/978-3-319-15986-6_1

2. Singh, P., Kumar, R., & Kumar, A. (2018). An automated and dynamic traffic signal control system using a multi-agent transport simulation approach. Procedia Computer Science, 132, 718-727. DOI: 10.1016/j.procs.2018.04.086

3. Kumar, A., & Kumar, R. (2021). Automated generation of traffic signals and lanes for MATSim based on OpenStreetMap. Journal of Ambient Intelligence and Humanized Computing, 12(6), 5655-5670. DOI: 10.1007/s12652-020-02647-3

4. Kumar, A., & Kumar, R. (2019). matsim-iitr. GitHub repository. Retrieved from https://github.com/teg-iitr/matsim-iitr

5. MATSim Community. (2021). MATSim User Guide. Retrieved from https://www.matsim.org/docs/

6. Anantharaman, R. (n.d.). MATSim Tutorial. Retrieved from https://noisemodelling.readthedocs.io/en/latest/Matsim_Tutorial.html