

UNIVERSITA' DEGLI STUDI DI BOLOGNA

School of Engineering and Architecture

International Master Course in Civil Engineering

Academic Year 2013/2014

Civil Engineering Research-A

Professor: Joerg Schweizer

Traffic Micro-simulation Modelling & SUMOPy User Mannual

Student:

Choudhry Muhammad Sharafat

Matricola: 0000657079

Email: muhammad.choudhry@studio.unibo.it

Abstract

The ever increasing world population, added to the always present need of human transportation, makes road traffic in the big cities a huge and urgent issue, and consequently, the target of substantial investigation. Traffic flow micro-simulations are interesting for transport planning problems due to their high temporal and spatial resolution. Unfortunately, most of them involve high computational costs making them impractical for running large scale scenarios. The study involves the SUMOPy overview which is intended to expand the user-base of the traffic micro-simulator SUMO (simulation of urban mobility) by providing a user friendly, yet flexible simulation suite. SUMOPy is discussed with all of its features by providing a user manual and a simple example is presented to show the use of the program. The applicability of the SUMOPy is presented to implement the traffic micro-simulation and the major steps involved to develop a modal to assess different outputs attained.

Contents

1. Introduction:.....	4
2. Future Urban Transport.....	5
3. Traffic Simulation Approaches	5
4. Micro-simulation:.....	6
5. General Characteristics of Micro-simulation Models.....	7
5.1 Time Steps.....	7
5.2 Randomization.....	7
5.3 Vehicle-trip Generation.....	8
5.4 Path Choice.....	8
5.5 Vehicle Movement Rules on Links.....	8
5.5.1 Single Vehicle on Link	8
5.5.2 Car Following Rule.....	9
5.5.3 Lane Changing Rule.....	9
5.6 Vehicle Movement Rules Within the Intersection (at the Node).....	9
6. Typical Micro-simulation Analysis Steps.....	9
7. SUMO Simulator:	10
8. SUMOPy User Manual:	11
8.1 SCENARIO:.....	11
8.1.1 Traffic Network.....	11
8.1.2 Zones.....	11
8.1.4 Trip & Route	12
8.1.5 Simulation Output.....	12
8.2 Network Creation:	14
8.2.1 Import Network:.....	15
8.2.2 Network Generation:.....	17
8.3 Demand Modelling:.....	17
8.3.1 Random Trips:.....	18
8.3.2 ZONES:.....	19
8.3.3 Origin-to-destination:.....	19
8.3.4 Routing by Turn Probabilities:.....	21
10 Example Demand/Routing by Turn Probabilities:.....	22
Conclusion:	26
References:.....	27

1. Introduction:

In transport planning, traffic flow simulation plays an important role as it takes the demand generated by an earlier process as input and produces derived quantities like traffic densities, flow volumes, speeds, and travel times as output. These quantities are in turn important for instance to analyze the state of the traffic system or to enable iterative demand generation based on one or multiple of these values.

Computational simulation has the advantage of allowing the assessment of system's behavior before it is developed or produced in real life. These simulations are the mathematical representation of the dynamic system, from which the conclusions of the properties from real system can be drawn. Time is the basic independent variable of a simulation model. In computer implementations of simulation models, the model state is updated at discrete times. The simulation model can either apply a time-based scanning approach in which the model is updated at regular intervals or an event-based strategy in which the model is updated at the points in time where the state of the system is changing. Event-based updating is less computer resource demanding as the simulation model is updated more sparsely than in a time-based model with equal accuracy. Event-based simulation does however imply calculation of the next change in the state of the model after each update. This procedure becomes very complicated for complex systems including many entities that change state frequently. Event-based simulation is consequently more appropriate for systems of limited size and for systems in which the entities change state infrequently. Time-based scanning is on the other hand considered to be appropriate for systems including large numbers of entities with frequently changing states.

Simulation models may also be either deterministic or stochastic. Deterministic simulation models do not include any randomness and are therefore appropriate for systems with little or no random variation. Stochastic simulation models make use of statistical distributions for some of the model parameters to reproduce the variability of the real system. The result of a model run of a stochastic model will consequently differ depending on the realization of the random numbers that are used to determine parameter values in the model.

Traffic simulation is the mathematical modeling of transportation systems (e.g., freeway junctions, arterial routes, roundabouts, downtown grid systems, etc.) through the application of computer software to better help plan, design and operate transportation systems. Traffic simulation models are designed to mimic the time evolving traffic operations in a road network. The simulator plays an important role as it takes the demand generated by an earlier process as input and produces derived quantities like traffic densities, flow volumes, speeds, and travel times as output. These quantities are in turn important for instance to analyze the state of the traffic system or to enable iterative demand generation based on one or multiple of these values. System complexity needs to be considered in order to provide the more appropriate simulation with practical feedback to real systems, time compression or expansion, higher control, and lower costs. These traffic simulations can study models too complicated for analytical or numerical treatment, can be used for experimental studies, can study detailed relations that might be lost in analytical or numerical treatment and can produce attractive visual demonstrations of present and future scenarios. To understand simulation, it is important to understand the concept of system state, which is a set of variables that contains enough information to describe the evolution of the system over time. System state can be either discrete or continuous. Traffic simulation models are classified according to discrete and continuous time, state, and space.

A Traffic simulation model consists of the representation of the road network together with the traffic in the network representing the supply and demand side of the traffic system respectively. The road network includes both the actual infrastructure and the traffic control systems. The traffic demand is commonly specified by an origin-destination matrix which specifies the number of trips between all origins and destinations in the traffic network during the time period that is to be simulated. Traffic simulation models are in general terms applied in studies of traffic conditions and the effects of traffic management strategies at the equilibrium between supply and demand in the traffic system.

2. Future Urban Transport

In the last century, since the economy became based on trading, the transport turned into an essential component of our society. These systems have become rather complex and extremely large, being geographically and functionally distributed, both in that respecting structure and management. Thus, first contemporary transportation revolution began with the introduction of electromagnetic communication, allowing greater integration through exchanging information more quickly and efficiently. Then, low-cost digital systems started playing an imperative role in transportation, improving efficiency of traffic control and coordination, as well as transport planning. Nevertheless, as transportation systems are becoming very large, both in terms of structure and dimension, the whole process of acquiring information from all sources, processing the essential data and providing adequate responses timely is rather a very arduous task.

By observing all FUT requirements, a question comes to our mind: How are we going to evaluate FUT systems? With the emergence of new paradigms, it is also needed to define new ways to measure if and how these requirements are being fulfilled. Current metrics, such as flow, time spent, storage utilization, and others, are not enough to totally evaluate these systems. Thus, as this new concept is usercentred, metrics also have to reflect user satisfaction (e.g. services availability, flexibility, and scalability). Also, many of the current measurements will be interpreted in a correlated form to extract system's nuances.

Artificial transportation systems *“is a generalization of the traffic simulation, which integrates the transportation system with other urban systems, such as logistic systems, social and economic systems, etc., to behave as a coordinate tool for transportation analysis, evaluation, decision-making and training.”*

Due to the high complexity and uncertainty of the transportation systems, traditional simulators are not able to capture the dynamics that characterize them. Persons can choose whether to travel or not, can change in any moment their planned routes, their choice can be affected by social or economic or environmental phenomena. Current transportation solutions are achieving high degree of autonomy and starts interacting with the user in a different dimension, as their peers. This turns the society into a system formed by multiple heterogeneous components with its own idiosyncrasy.

Simulation is a key component in this new step of mobility systems, due to the increased complexity in the test and validation task, which is especially more byzantine due to real-time constraints and the presence of heterogeneous participating entities (vehicles, urban infrastructures, traffic infrastructures, pedestrians etc.). In our view the new platform must support, natively or by extension, distributed and autonomous decision-making capabilities, that is the Multi-Agent paradigm, different types of communication techniques, it should simulate various types of heterogeneous entities providing as realistic as possible easy results and last should offer on-line simulation visualization, in order the user can extract the desired information.

3. Traffic Simulation Approaches

Traffic simulation is largely studied and can be classified in four types: macroscopic, mesoscopic, microscopic and nanoscopic.

- **Macroscopic**

Macroscopic simulation models the flow of traffic using high-level mathematical models often derived from fluid dynamics, thus it is continuous simulations. This type of simulation handles every vehicle in the same way and as group. It uses aggregate input and output variables such as speed, flow and density. Macroscopic simulators are most useful for the simulation of wide-area traffic systems, which do not require detailed modelling, such as motorway networks and interregional road networks. This approach is not very realistic because in real life there are many different types of vehicle driven by different individuals who have their own styles and behaviours. However, it is fast and accurate but is not well suited to urban models in general.

- **Microscopic**

Microscopic simulators model individual entities (e.g. vehicle, driver etc.) separately at a high level of detail, and are classified as discrete simulations. Here, interactions are usually governed by the car-following and lane-changing logics. Thus, traffic flow details, usually associated with macroscopic simulation are the emergent properties of the microscopic simulation. These simulators can model traffic flow more realistically than

macroscopic simulators do, due to the extra details added in modelling the simulated entities individually. Microscopic simulators are widely used to evaluate new traffic control and management technologies as well as performing analysis of existing traffic operations.

- **Mesosopic**

mesoscopic simulators fill the gap between macro and micro simulators. They normally describe traffic entities at a higher level of detail, than macroscopic models but their behaviours and interactions are in a lower level of detail. In mesoscopic model, vehicles can be grouped in packets, which are routed throughout the network and are treated as one entity. Other paradigm is that of individual vehicles that are grouped into cells to control their behaviour. The cells traverse the link and vehicles can enter and leave cells when needed, but not overtake.

- **Nanosopic**

A new trend of traffic simulation is the nanoscopic model that extends the vehicle vision, dividing it in parts. It is mostly used in autonomous driving and is in a strict relationship with automated robotic, because of the need to simulate sensors. Figueiredo et al [3] observed a great potential use of robotic simulators in the autonomous driving field, motivating an information exchange among robotic and traffic study groups.

4. Micro-simulation:

Micro-simulation is the dynamic and stochastic modeling of individual vehicle movements within a system of transportation facilities. Each entity (e.g. vehicle, driver etc.) is modeled separately at a high level of detail, and are classified as discrete simulations. Each vehicle is moved through the network of transportation facilities on a split second by split second basis according to the physical characteristics of the vehicle (length, maximum acceleration rate, etc.), the fundamental rules of motion (e.g. acceleration times time equals velocity, velocity times time equals distance) and interactions are usually governed by rules of driver behavior (car following and lane changing rules, etc.). Traffic micro-simulation models are becoming increasingly popular for the evaluation and development of road traffic management and control systems.

These models provide a better and purer representation of the actual driver behavior and network performance. They are the only modeling tools available with the capability to examine certain complex traffic problems (e.g. intelligent transportation systems, complex junctions, shock waves, effects of incidents. The powerful graphics shows visual representation of individual vehicle traversing across networks that include a variety of road categories and junction types. This visual representation could be very useful to gain more widespread acceptance of complex strategies. The micro-simulation can be used to develop new systems and optimise their effectiveness. They can easily estimate the impacts of a new scheme by producing outputs on a wide range of measures of effectiveness.

Micro-simulation is particularly suited to the development, testing and evaluation of intelligent transportation systems (ITS). Many such systems interact with individual vehicles. Responsive signal control, public transport priority and ramp metering systems react to vehicles approaching junctions. Dynamic Route Guidance systems supply specific information to individually equipped vehicles. Intelligent Cruise Control systems adjust the speeds of equipped vehicles. Therefore to assess the potential benefits of using ITS, it makes sense to use an assessment tool that is capable of modeling interactions at the level of individual vehicles. Micro-simulation models, which can reproduce individual driver behaviour, should therefore be an essential part of any such assessment tool.

Moreover, as individual vehicles are being modeled it is often possible to use the micro-simulator as a proxy for the real world and connect it to real systems directly. This negates the need to produce a model of the system being assessed.

5. General Characteristics of Micro-simulation Models

The typical micro-simulation software implements a combination of procedures (models) for identifying the location, speed, and acceleration rate of vehicles in the highway network at each moment of time. A set of relatively simple rules is used to move the vehicles through the network. Statistics are tabulated on the vehicle activity and two outputs are typically produced: text reports, and visual animations.

Basic elements of micro-simulation:

1. **Edges and Lanes:** The edge is a plain part of a road network, like a highway or a normal street which connects two roads and each edge consist of number of lanes mostly two lanes for traffic flow in different directions defining the maximum speed allowed, length and geometry of the lane.
2. **Nodes:** Nodes are the junctions which connects the edges.
3. **Vehicle/persons:** Vehicle is a mode of transport used for the traffic flow in the network for the simulation and it is defined by the route the vehicle shall drive along, vehicle colour, time step at which the vehicle shall enter or leave the network, lane on which the vehicle shall be inserted or leave the network, position at which the vehicle shall enter or leave the network and speed of vehicle.
The person checks whether a vehicle with a line from the given list is stopping at the given edge. If so, it enters the vehicle and starts the ride. It does not check whether the vehicle has the aspired destination on the current route. The first time the vehicle stops (on a well defined stop) at the destination edge, the ride is finished and the person proceeds with the next step in the plan.

5.1 Time Steps

The simulation of vehicle movements is done in a series of time steps. The vehicle position, velocity, and rate of acceleration/deceleration are computed at the end of each time step and statistics accumulated on the results. The number of time steps per second influences both the accuracy and duration of a micro-simulation model run. The more time steps per second, the greater the potential precision of the model results. A model that computes the vehicle position only once every second will not be as precise as a model that has several time steps per second.

This precision becomes more important for freeway analysis, when vehicles travel at higher speeds. A vehicle moving at 30 mph on a city street travels 44 feet (2.5 car lengths) in one second. A vehicle moving at 60 mph on a freeway moves 88 feet (5 car lengths) in one second.

If the time step resolution is only 1 time step per second, then a vehicle moving at 60 mph cannot safely follow another vehicle by closer than 5 car lengths and avoid a collision if the lead vehicle decides to slow down between one time step and the next.

However, the more steps there are per second the more computations required to perform a given time period of simulation. Doubling the number of time steps per second doubles the computations.

It is also possible to go too far in the search for increasing precision. Some driver behavior models require knowledge of historical conditions (what was the lead vehicle's speed 3 time steps before the current time step) to assess drivers' reactions. If the analyst makes the time steps exceedingly short, this may bias the computations.

Once the model has been calibrated, the number of time steps per second should not be changed. Changing the time steps per second will affect the simulation results.

In general, vehicle trips and routes can be either generated before the actual simulation (using different demand models) or during the simulation (agent based simulation). In this document we follow a process where vehicle trips and routes are generated before the micro-simulation is launched.

5.2 Randomization

Micro-simulation models would produce unrealistically regimented simulations with all drivers moving at the same time and in the same way, if it were not for randomization. The simple rules used to move vehicles in a micro-simulation do not realistically reproduce the wide range of human behavior observed in the real world. Random variables are used to produce a plausible range of human behavior from the simple rules.

Computer software uses a random number generator to generate the necessary set of random variables. The generator requires a starting number, or “seed” to produce a unique sequence of numbers. The same seed, used with the same generating routine, on the same computer will produce the same sequence of numbers for use in the random variables, every time.

Thus, a single micro-simulation model run is like rolling the dice only once. In order to find out the average conditions it is necessary to operate the micro-simulation model several times, with different random number seeds and then average the results of the different runs. A discussion of the required number of runs to estimate mean and 95 percentile results is provided in the Alternatives Analysis chapter.

5.3 Vehicle-trip Generation

Trip generation is done before actual micro-simulation and each trip is defined by the depart time, vehicle type, origin/destination link, origin/destination position on link.

At the start of every time step in the simulation, the micro-simulation software makes a decision about whether or not to release one or more vehicles onto the road system. Vehicles may be released from specified entry and exit points on the network and releasing vehicles in zones is also done by releasing vehicles on a link (which is within a defined zone, see make taz in section 9.3.2). The decision to release is a random decision with an expected value specified by the mean trip generation rate (vehicles/time step) for the zone. The random number generator is used to decide whether or not a vehicle is released. Note that this process does not guarantee that the final number of vehicles released onto the network will actually match the value coded by the analyst. The average will be close to the input value, but not identical.

At the time the vehicle is generated it is randomly assigned a vehicle type (which may parameterize also the driver with specific parameters such as aggressivity). The vehicle type determines its length, weight, width, height, maximum speed, maximum acceleration rate and the braking rates.

If the software allows for the input of vehicle origin-destination tables, the vehicle will also be assigned a destination, also using a random number generator. The probability of a vehicle being assigned a particular destination is specified by the origin-destination table.

5.4 Path Choice

If the software provides for the input of origin-destination tables, then a path (series of connected links leading to the assigned destination) will be computed. This path can be updated dynamically during the simulation run as the vehicle moves along the network and congestion conditions change on the network.

Various methods are available for determining the path chosen by the vehicle including: Shortest Path, Stochastic, or Dynamic4.

If the software does not use an origin-destination table then the vehicle’s ultimate destination and path will be determined as it moves down the network according to the intersection turn probabilities coded by the analyst. The vehicle will be randomly assigned to each turn movement at each intersection, making it possible for individual vehicles to drive in circles, unless the user over-rides this option.

5.5 Vehicle Movement Rules on Links

This is the actual micro-simulation. During the micro-simulation, the vehicles enter the simulations at the previously specified times and positions and start to move along their route, according to certain rules which are designed to mimic real vehicle behavior.

A link is a section of street where street geometry and demand are sufficiently constant so that the section of street can be modeled as a pipe, with vehicles entering the pipe at one end and leaving at the other end. There are three basic vehicle movement rules on a link: single vehicle, car following, and lane changing. For most all models, the specifics of vehicle behavior will vary a bit between freeway links and non-freeway links. Driver awareness and aggressiveness will modify vehicle behavior as well.

5.5.1 Single Vehicle on Link

In the absence of any other vehicles, the micro-simulation software will move the vehicle down the link at its desired speed. The desired speed of the vehicle is determined by the analyst coded free-flow speed for the link

and the driver type (aggressiveness). More aggressive drivers may go faster than the coded free-flow speed, less aggressive drivers may go slower. Some software may insert random accelerations and decelerations into the vehicle speed as it travels down the link. Others will leave the speed fixed for the length of the link, in the absence of other vehicles or obstructions in the downstream link.

5.5.2 Car Following Rule

When a vehicle catches up to another vehicle on the link, a car following rule and a lane changing rule are used to decide how the vehicle will respond to the lead vehicle. The software will employ a mean target following headway which may be varied according to driver aggressiveness and vehicle type. The definition of headway does vary by software. All software generally set their car-following rules to guarantee that vehicles will not collide.

5.5.3 Lane Changing Rule

Lane changing requires that an acceptable gap be available in the traffic stream in the adjacent lane. The lane changing rules will vary according to the situation. Mandatory lane changes are made when the road geometry dictates the change (such as a designated turn lane, or a lane drop). Discretionary lane changes are made to pass slower moving vehicles. The probability of a vehicle making a discretionary lane change is usually a function of the difference in speed between the two vehicles and the available gaps between vehicles in the adjacent lane. Anticipatory lane changes may be made to pre-position the vehicle for an upcoming turn, or to avoid slower vehicles entering the freeway on an on-ramp. The upstream decision point where vehicles start to make anticipatory lane changes (sometimes called the signpost or warning sign location) is often difficult to model realistically in today's simulation software. Some software provide for the vehicle to look ahead a set number of links (which can become a problem when numerous short links are coded). Other software may employ "path-based" pre-positioning that is not constrained by the number of links.

5.6 Vehicle Movement Rules Within the Intersection (at the Node)

Gap acceptance is used to move left turning vehicles (with permitted phasing) through opposing through vehicles, or stop sign controlled vehicles through the other vehicle streams. Each vehicle movement at an intersection is assigned a priority ranking. Lower priority vehicle streams defer to higher priority movements. No crosscheck is typically made by most simulation software of vehicle movement inside of the intersection (once it has accepted a gap) to ensure that two vehicles do not occupy the same space at the same point in time (an apparent collision). If the analyst were to code two conflicting green phases, the simulation software may well show two streams of traffic driving through each other. The vehicle speed may be slowed inside an intersection while it is making a turn. The reduced speed may be fixed by turn movement type, or, in some software, it may vary according to the radius and angle of the turn.

6. Typical Micro-simulation Analysis Steps

The major steps involved in a micro-simulation analysis are:

1. Identification of Project Purpose, Scope, and Approach

This scoping step is critical for determining the ultimate cost and schedule for the micro-simulation analysis.

2. Data Collection

This step involves the collection of input data for the micro-simulation model as well selected output data for calibrating the model.

3. Demand Generation

The model coding step is where the analyst converts the field data into inputs for the micro-simulation model (trips and finally routes)

4. Error Checking

The error-checking step verifies the accuracy of the coded input data.

5. Calibration

Calibration is where the analyst adjusts the default parameters in the standard behavioral models contained in the micro-simulation software to local conditions.

6. Alternatives Testing

This step is the purpose for which the micro-simulation model was developed.

7. Documentation

Documentation provides information on the inputs to the model, the validity of the model, and the results of the alternatives analysis.

8. Presentation of the Results

This step is where the analyst presents the micro-simulation analysis results to decision makers and the general public.

7. SUMO Simulator:

"Simulation of Urban MObility", or "SUMO" for short, is an open source, microscopic, multi-modal traffic simulation designed to handle large road networks. Its features include: collision free vehicle movement, multi-lane streets with lane changing, fast execution speed, dynamic user assignment, and other. It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route, and moves individually through the network. The applications of SUMO included:

- traffic lights evaluation
- route choice and re-routing
- evaluation of traffic surveillance methods
- simulation of vehicular communications
- traffic forecast

SUMO uses the TraCI layer to control it through TCP connections. However, it has limited functions to control the simulation process and requires that information from the simulator be gathered through sensors spread out over the network. The simulator is developed in the Institute of Transportation Systems at the German Aerospace Center. It is licensed under the GPL.

8. SUMOPy User Manual:

In this report concentration is exerted on the SUMOPy which is intended to expand the user-base of the traffic micro-simulator SUMO (simulation of urban mobility) by providing a user friendly, yet flexible simulation suite. A further scope of SUMOPy is to manage the huge amount of data necessary to run complex multi-modal simulations. It includes different demand generation modals as well as a large range of modes such as public transport, bicycle and Personal Rapid Transit (PRT). SUMOPy consists of a GUI interface, network editor as well as simple to use scripting language which facilitates the use of SUMO. The simulator is chosen since it can support the characteristics for FUT requirement.

8.1 SCENARIO:

The term scenario in SUMOPy describes the micro-simulation of traffic of particular event to be carried out or the series of actions involved.

8.1.1. Traffic Network:

Transport planning uses mathematical transport network model (a graph) to predict the performance and impacts of present and future transport networks. This model will be the bases for all subsequent analyses, including transport demand estimation and traffic assignment (determination of vehicle flows on each network link). Once the vehicle flows on the entire network are known, we can quantify its environmental, economical and social impact. A transport network is made of links and nodes. The links are actually modeling the physical transport line (road, railway, footpath), while the nodes are points where links can be joined (All of this network information is contained in *scenario.net.xml* file).

8.1.2. Zones:

The first step in the network extraction is to define the boundaries of the study area. It is important to define the study area such that it covers the activity-centers of potential users of a planned transport project. The spacial division of the study area in zones is called *zoning*. With zoning, the origins as well as the destinations of trips (during a defined time interval) within a zone are lumped together. Zoning has two main purposes:

- i. zones allow to simplify the modeling of the transport network and
- ii. they allow to apply statistical methods to estimate the transport demand.

The information is contained in the Traffic Assignment Zone(TAZ) file which is *scenario.taz.xml*

8.1.3. Flows & Turns:

Network flows are the movement of objects through the network and turns are added to the network dataset for one of two reasons: to restrict travel, such as no left turn allowed; or to add an additional travel time for the solver to perform that maneuver (Flow and turn flow information exists in *scenario.turnflow.xml* file).

8.1.4. Trip & Route:

A **trip** is a vehicle movement from one place to another and, it define the start and end point of a movement with optional changes in mode. It contains the information of:

- The edge at which the trip starts.
- The destination edge at which the trip ends.
- Vehicle type (described in section 8.3.1 in Random trips).

The information of the trip file is contained in *scenario.trip.xml* file.

A route is connecting the link of origin and destination with a sequence of links contains the information of :

- A vehicle type which describes the vehicle's physical properties contains the information of the vehicle type, vehicle length, maximum speed.
- A route the vehicle shall take (The edges the vehicle shall drive along, given as their ids, separated using spaces).

The information of the route file is contained in the *scenario.rou.xml* file.

8.1.5. Simulation Output:

Output is the simulation result. Most of micro-simulation models use a Graphical User Interface. It is generally an on-line animation with which user can visualise vehicle movements and state of traffic and signals, display various traffic variables and path information by clicking on objects (vehicles, links, etc.) and have an overview of traffic conditions by zooming capability and by, for example, colouring links according to density and velocity. SUMO allows to generate a large number of different measures (information of the simulation output is contained in different respective *result files*).

Some of the available outputs are:

- **Raw vehicle positions dump:**

The network dump is a xml-file containing for each time step every edge of the network with every lane of this edge with all vehicles on this lane. For each vehicle, its name, speed and position on its respective lane are written

- **Trip information:**

This output contains the information about each vehicle's departure time, the time the vehicle wanted to start at (which may be lower than the real departure time) and the time the vehicle has arrived. The information is generated for each vehicle as soon as the vehicle arrived at its destination and is removed from the network.

- **Vehicle routes information:**

The vehicle routes output contains information about which route a vehicle took and if his route was replaced at any time by a new one, each of the previous routes together with the edge at the time their replacement took place is reported. Furthermore, the times the vehicle has entered and has left the network are stored herein.

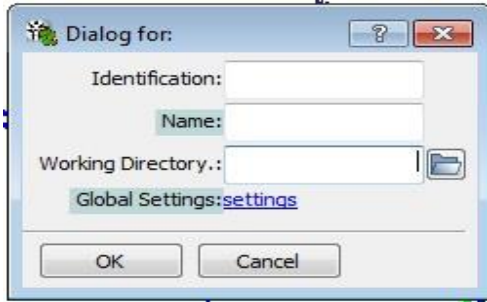
- **Simulation state statistics:**

This output contains the simulation-wide number of vehicles that are loaded, inserted, running, waiting to be inserted, have reached their destination and how long they needed to finish the route. The last value is normalised over all vehicles that have reached their destination so far.

In order to create a scenario, the scenario-menu have to be selected with the following items:

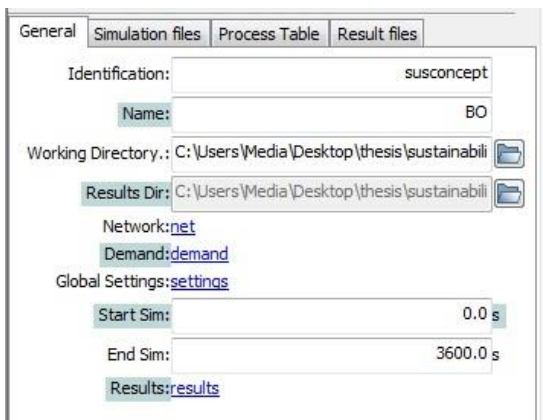


The window appears on selecting the 'Create' to create a scenario is;



'Working directory' is a folder created for a particular scenario which would contain all of its files being created during the process. 'Name' could be chosen as any random name and the 'identification' is the name of the specific scenario. Further the Name can contain any sequence of characters, whereas identification should be without spaces, commas, accents, mathematical operators, forward and backward slashes.

The features inside the control panel are shown below which lies on the left side window in program and it shows the parameters and the data given for the relevant features like for 'scenario' it is;



The time for starting simulation is set to 0.0 seconds and for ending set to 3600.0 seconds by default which could be changed.

Simulation files:

- Net files; It contains all of the network information in *scenario.net.xml* file
- Poly files, contains information of polynomials with buildings and other places (territory) in *scenario.poly.xml* file
- Zones; Zoning information is contained in the Traffic Assignment Zone(TAZ) file which is *scenario.taz.xml* (zoning and make taz is discussed briefly in section 8.3.2).
- ODMs; It contains all origin to destination information in *scenario.odm.xml* file (demand by ODM is described in section 8.3.3)
- Flows and turns; It represents flow and turn flow information in *scenario.turnflow.xml* file (routing by turn probabilities is described in section 8.3.4)
- Trips; contains the information of trips in *scenario.trip.xml* file.
- Routes, all the routes information is contained in the *scenario.rou.xml* file.

Result files:

- edge emissions (information on edge emissions of the completed trips in *scenario.out.edgeemissions.xml* file);
- edge noise (information on edge noise of completed trips in *scenario.out.edgenoise.xml* file);
- vehicle route (information on route traffic of completed trips in *scenario.out.vehroute.xml* file)
- trip information (information on trip traffic in *scenario.out.tripinfo.xml* file)
- edge data route (information on edge data of completed trips in *scenario.out.edgedata.xml* file).

*The simulation files and result files are generated during micro-simulation process and can be observed at the end of simulation with all of the values.

8.2 Network Creation:

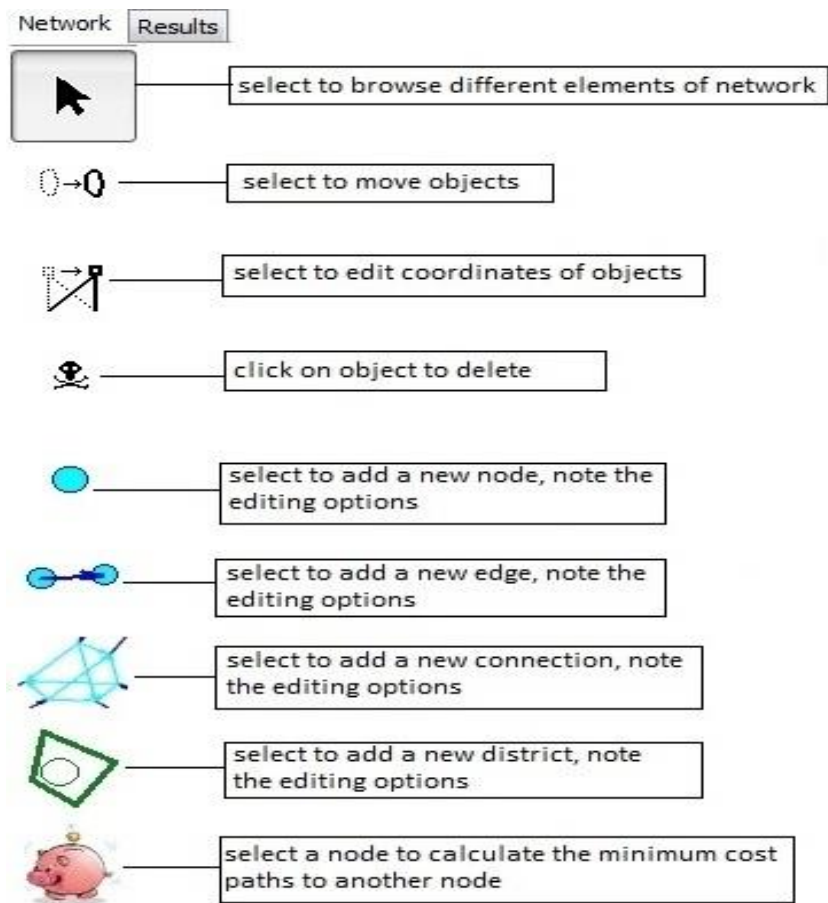
A **SUMO network file** describes the traffic-related part of a map, the roads and intersections the simulated vehicles run along or across. At a coarse scale, a SUMO network is a directed graph. Nodes, usually named "junctions" in SUMO-context, represent intersections, and "edges" roads or streets. Note that edges are unidirectional. Specifically, the SUMO network contains the following information:

- every street (edge) as a collection of lanes, including the position, shape and speed limit of every lane,
- traffic light logics referenced by junctions,
- junctions, including their right of way regulation,
- connections between lanes at junctions (nodes).

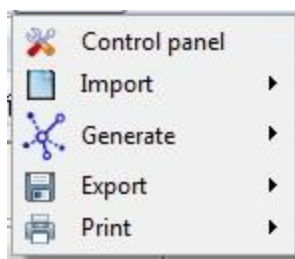
Also, depending on the used input formats and set processing options, one can also find

- districts,
- roundabout descriptions.

There are different options exist in the window of Network in SUMOPy which could be selected for respective features to edit the traffic network and are as follows;



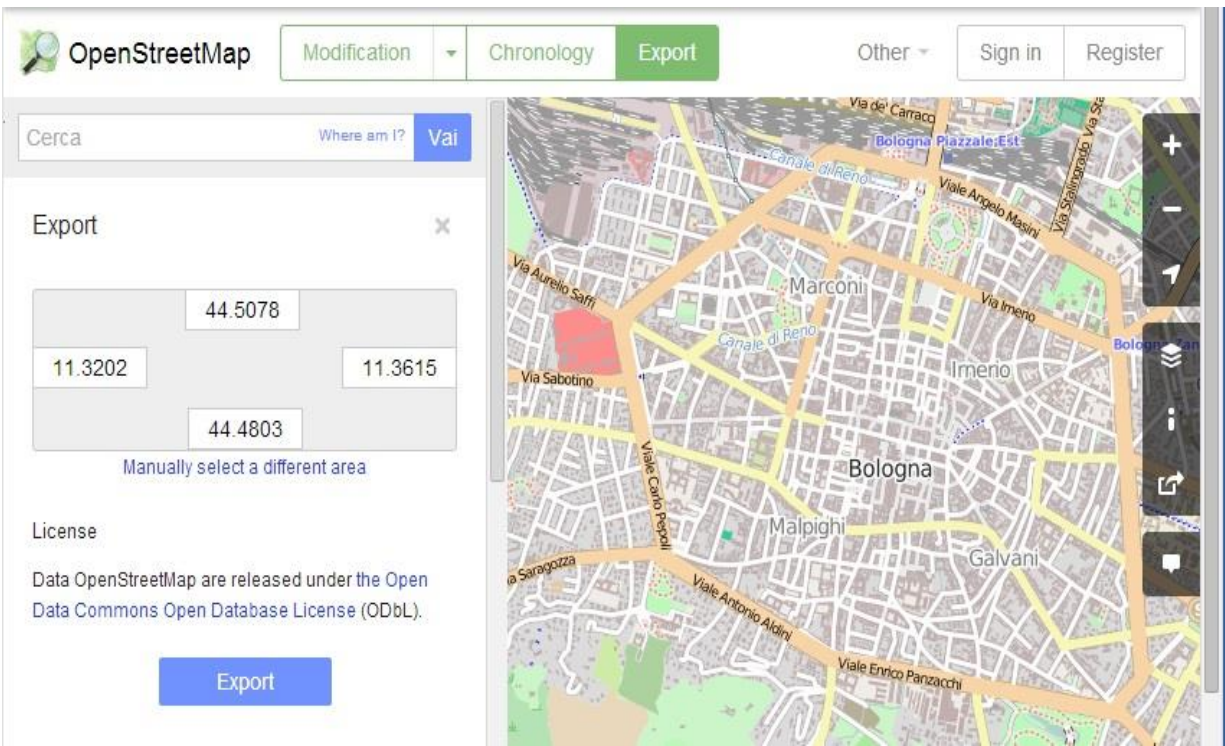
On selecting the Network in SUMOPy, following options will appear;



8.2.1 Import Network:

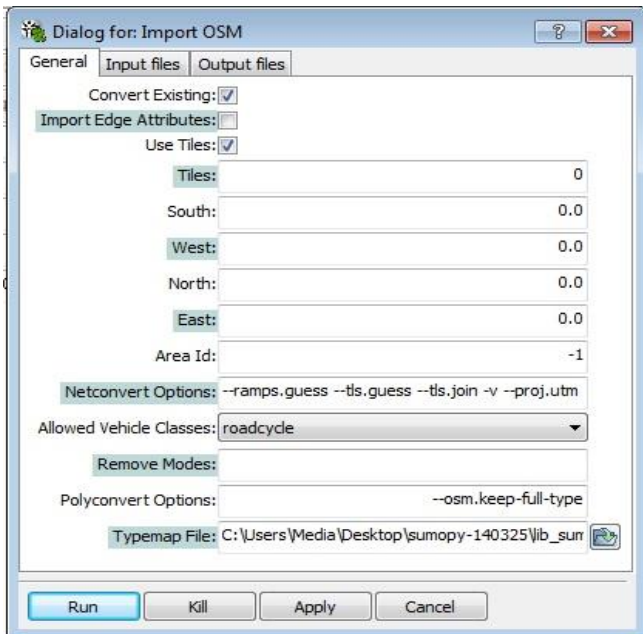
On SUMOPy, it is also possible to import the network from open street map (OSM) by selecting the specific area choosing the coordinates by zooming and inserting these coordinates on SUMOPy in the required fields. (by choosing *Network>Import>osm*).

OSM website: (<http://www.openstreetmap.org/export#map=16/44.4972/11.3593>).



On inserting the coordinates, SUMOPy extracts the network from OSM itself with the required traffic information for simulation. Very large areas could be retrieved from the open street map by bounding box using the option 'Tiles'. Tiles forms the osm map and contains the data in the added layers such as traffic information. Tiles are taken automatically itself while the specific area is retrieved from the map depending on the area size.

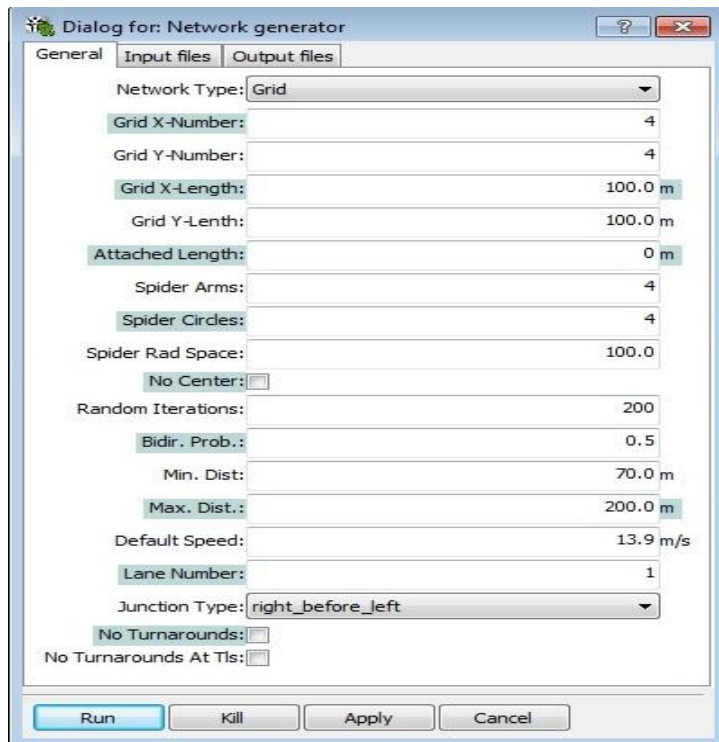
It is possible to import the network by choosing *Network>Import>osm*, which will appear as;



8.2.2 Network Generation:

The new network can also be generated on the SUMOPy (by choosing *Network>Gnerate>Gnerate network*) and there are three different kind of networks exist in the program 'Grid', 'Spider' and 'Random'.

For each kind of network, data is set by default for different options as shown below but this could be changed as per requirement.

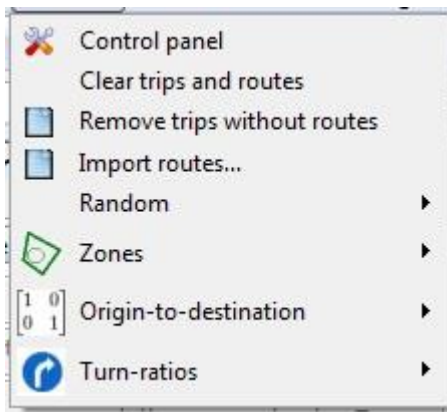


After generating the network, different parameters used and the data utilized could be observed in the control panel. This information exists in the tables in control panel could be further changed and modified as per requirement.

8.3 Demand Modelling:

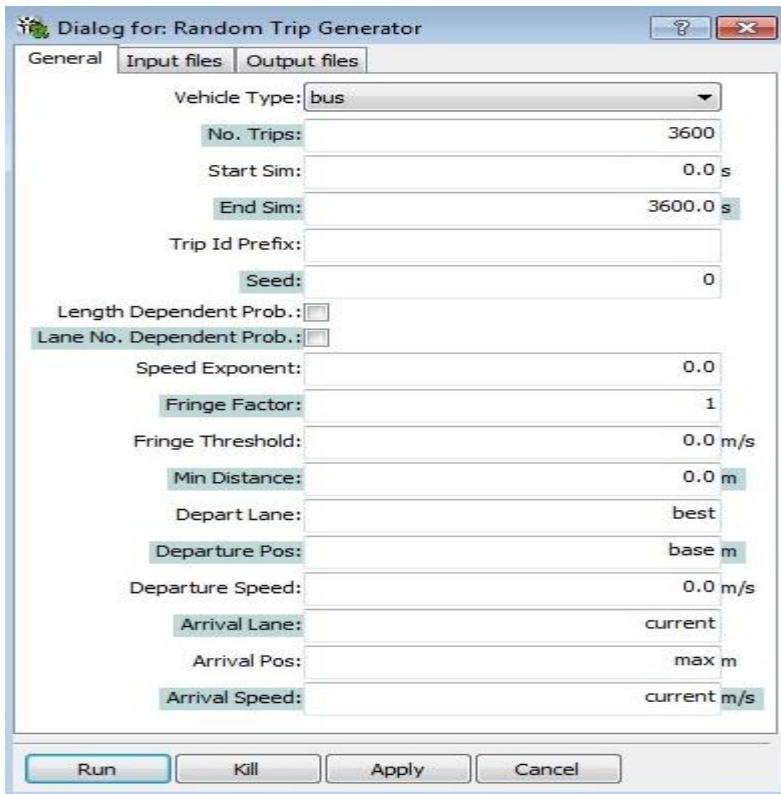
After having generated a network, one could take a look at it using **SUMOPy**, but no cars would be driving around. One still needs some kind of description about the vehicles. This is called the **traffic demand**, for which the following nomenclature: A **trip** is a vehicle movement from one place to another defined by the starting edge (street), the destination edge, and the departure time. A **route** is an expanded trip, that means, that a route definition contains not only the first and the last edge, but all edges the vehicle will pass. **SUMO** and **SUMO-GUI** need routes as input for vehicle movements. There are several ways to generate routes for SUMO.

- **Random Trip Generation** (a quick way to get some traffic if you do not have access to any measurements but the results are highly unrealistic)
- **Origin to Destination** (Origin-Destination-Matrices (or OD-matrices) are often available from traffic authorities. They have to be converted to trips using **Odm to trips**)
- **Turn ratios** (One may also leave out the destination edges for flows and use turning ratios at junctions instead)



8.3.1 Random Trips:

Random generates the set of random trips for a given network and it predicts the number of trips originating in or destined for a particular traffic analysis zone. It does so by choosing source and destination edge either uniformly at random or weighted by length, by number of lanes or both. The trips are distributed evenly in an interval defined by begin (option "Start sim", default 0) and end time (option "End sim", default 3600) in seconds. The number of trips is defined by the repetition rate (option "-p", default 1) in seconds. Every trip has an id consisting of a prefix (option "-t", default "t") and a running number.



The script does not check whether the chosen destination may be reached from the source. This task is performed by the router.

min-distance ensures a minimum straight-line distance (in meter) between start and end edges of a trip. The script will keep sampling from the edge distribution until enough trips with sufficient distance are found.

fringe-factor increases the probability that trips will start/end at the fringe of the network. If the value *10* is given, edges that have no successor or no predecessor will be 10 times more likely to be chosen as start- or endpoint of a trip. This is useful when modelling through-traffic which starts and ends at the outside of the simulated area.

Some parameters are set as default but all the parameters could be inserted depending on the scenario.

Vehicle type is used to choose the type of vehicle to generate the random trip in the program and is set by default as:

- Bus
- Passenger
- Bicycle
- Motorcycle
- PRT

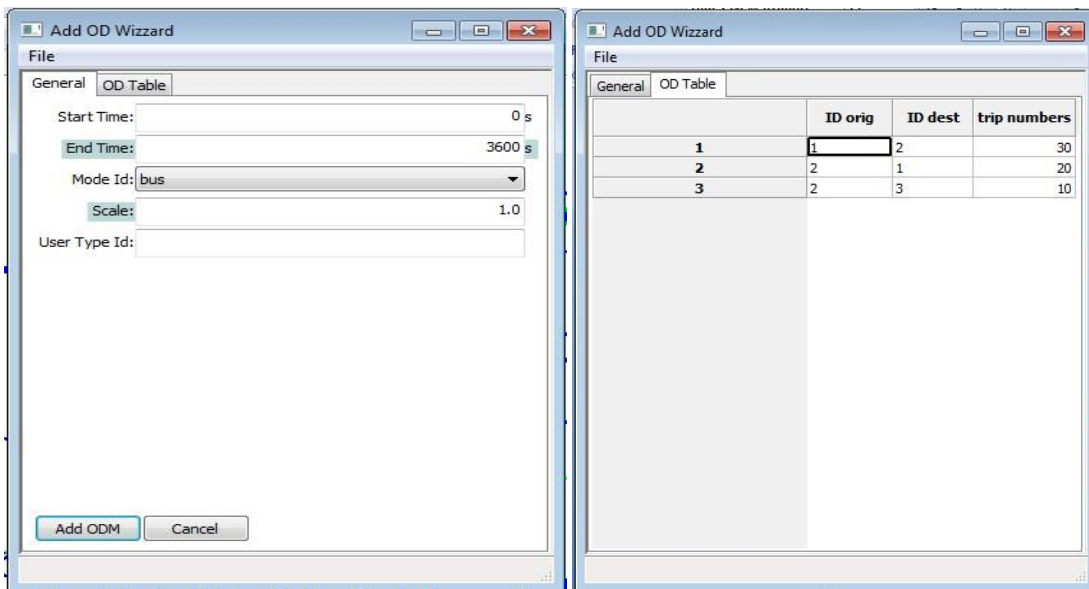
8.3.2 ZONES:

Districts are created on the given network which are the origin and destinations of the flow to define the traffic demand by using ODM or turn-ratios patterns. After creating the zones(districts), are saved as .taz file which exists with other files of the scenario . So this file could be import by choosing (*demand>zones>Import taz*).

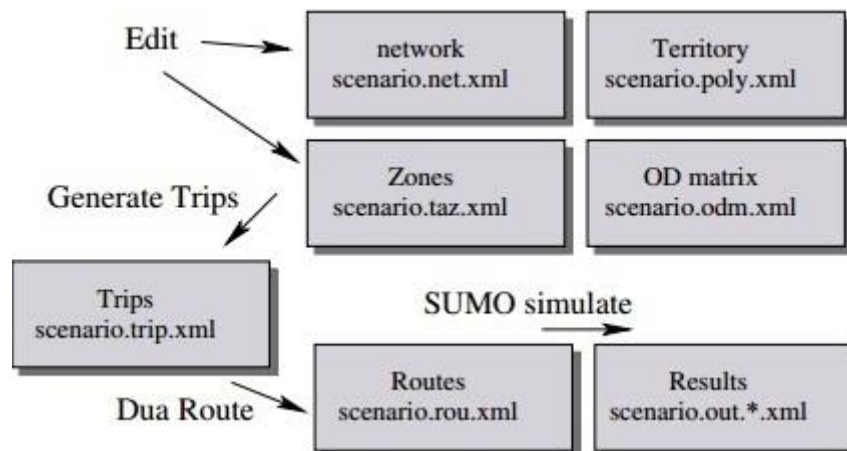
Make taz (demand>zones>make taz) use zones (districts) as routing end points. It is described by its id, being a simple name, and lists of source and destination edges. A TAZ should have at least one source and one destination edge, each described by its id and use probability called weight herein. These edges are used to insert and remove vehicles into/from the network respectively. The probability sums of each the source and the destination lists are normalized after loading.

8.3.3 Origin-to-destination:

It can be used to insert the traffic demand through OD Table which are estimated by the traffic counts. OD Table is created by inserting the set of origin '*ID orig*', destination '*ID dest*' and the number of trips '*trip numbers*' and this information can be inserted in OD table for different kind of traffic modes (option '*Mode id*') present by default in the program.



ODM-TRIPS computes trip tables from O/D (origin/destination) matrices. It assumes the matrix / the matrices to be coded as amounts of vehicles that drive from one district or traffic assignment zone (TAZ) to another within a certain time period. Because the generated trips must start and end at edges, **ODM-TRIPS** requires a mapping of TAZ to edges.



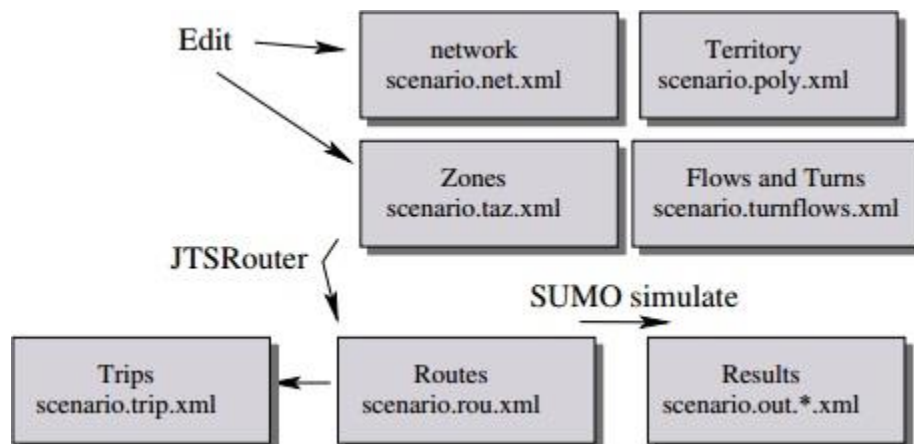
TRIPS-ROUTES determine the routes from the trips which contains not only the first and the last edge, but all edges the vehicle will pass.

8.3.4 Routing by Turn Probabilities:

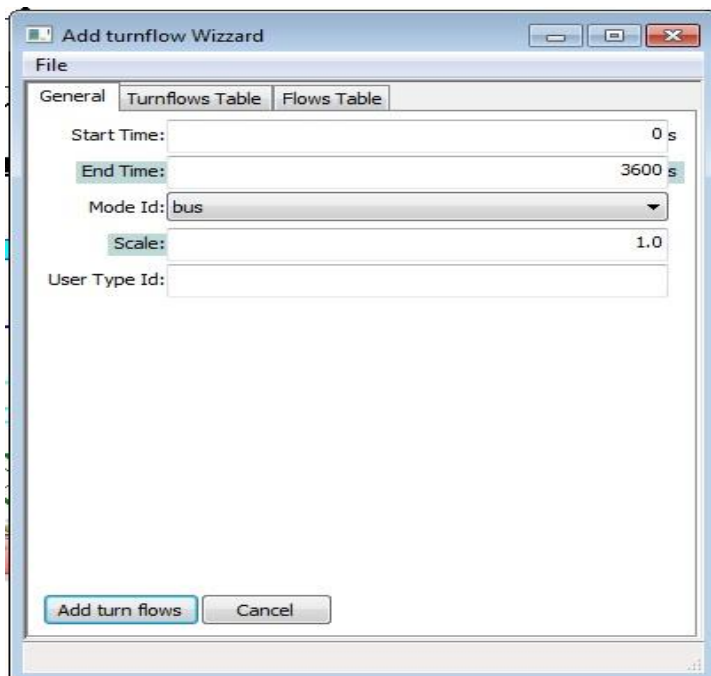
Turn-ratios, in this kind of pattern for each interval and each edge, turn flows have to be assigned with chosen traffic mode and time duration.

Jtrrouter is a routing applications (chosen by *demand>turn-ratios>jtrroute*) which uses flows and turn flows (absolute flows) at junctions as input by utilizing the following parameters:

- the network to route the vehicles through,
- the description of the turning ratios for the junctions (defaults may be used for this, too), and
- the descriptions of the flows.



For each edge, **source** (the edge from which vehicle entering the network), **sink** (vehicle leaves the network as soon as it comes to a sink edge) and **turnflows** (percentage of vehicle entering the edge) have to be inserted in the turnflows table for a specific transport mode (option ‘Mode id’), i.e.



Mode Id contains the type of transport modes which are:

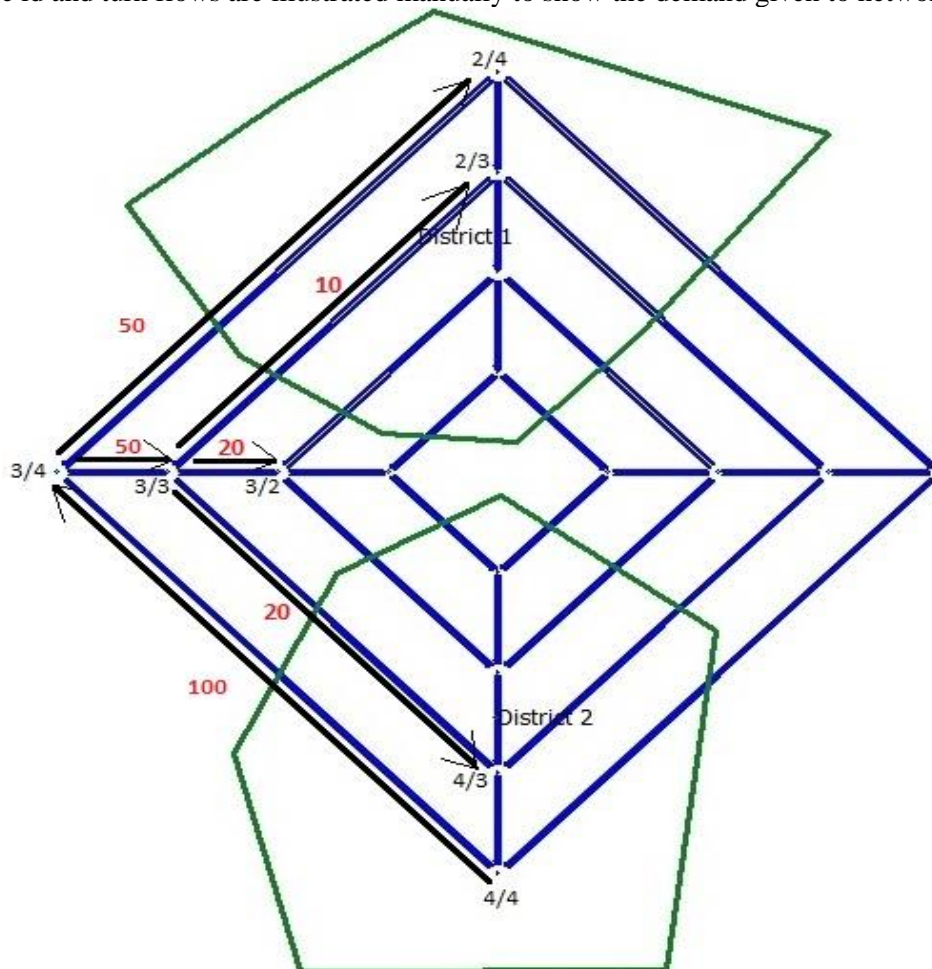
- Bus
- Bicycle
- Motorcycle
- Private
- Public Transport

It is also possible to create a turn-ratio demand on notepad containing edge ids and turn flows, and imported by choosing 'file' from 'Add turn flow wizard (*demand>turn-ratios>add*)'

10 Example Demand/Routing by Turn Probabilities:

Turn probabilities derived from traffic counts can be used as an alternative to origin-destination matrices as inputs to traffic simulation. For small scale traffic simulation (10 to 100 intersections), it may not even be feasible to collect OD matrix data.

Here is a small example of routing by turn-ratios implemented by choosing the Spider type network in SUMOPy. Two zones (districts) are generated on the network, the traffic network is represented below (black arrow lines, node id and turn flows are illustrated manually to show the demand given to network)



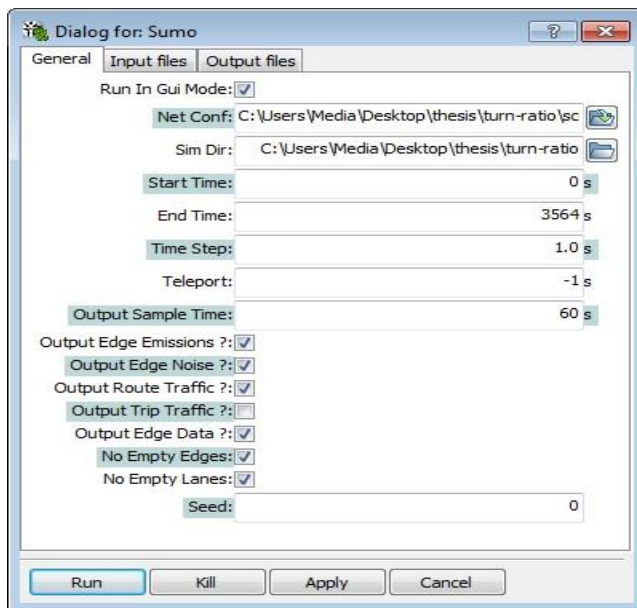
Once the network is developed on the SUMOPy, the traffic demand is given by using the turn-ratio for which only the traffic mode 'bus' is selected and the bus flow is taken as 100 and turn flows given to the network links is as follows:

```
4/4to3/4,100,3/4to3/3,50,3/4to2/4,50
3/4to3/3,0,3/3to4/3,20,3/3to2/3,10,3/3to3/2,20
```

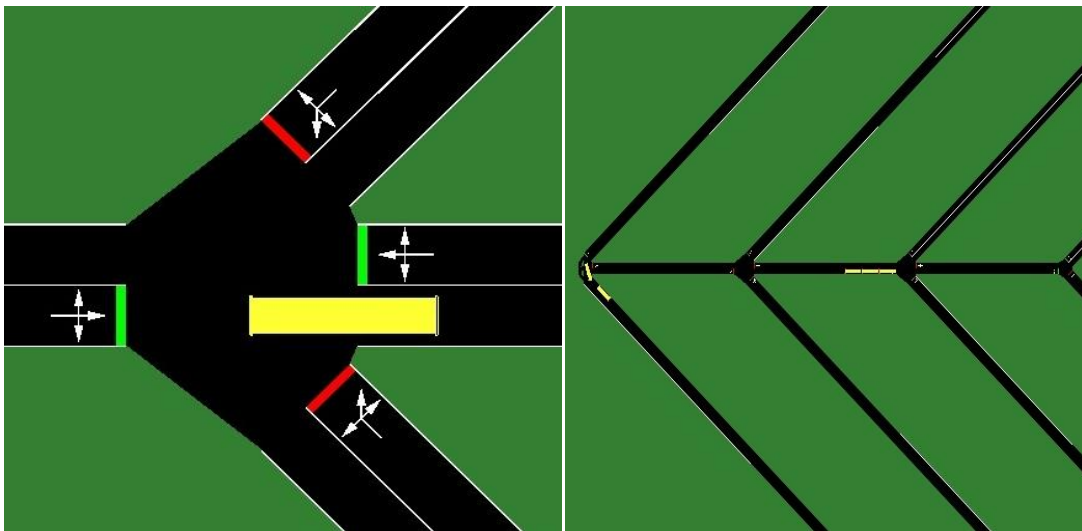
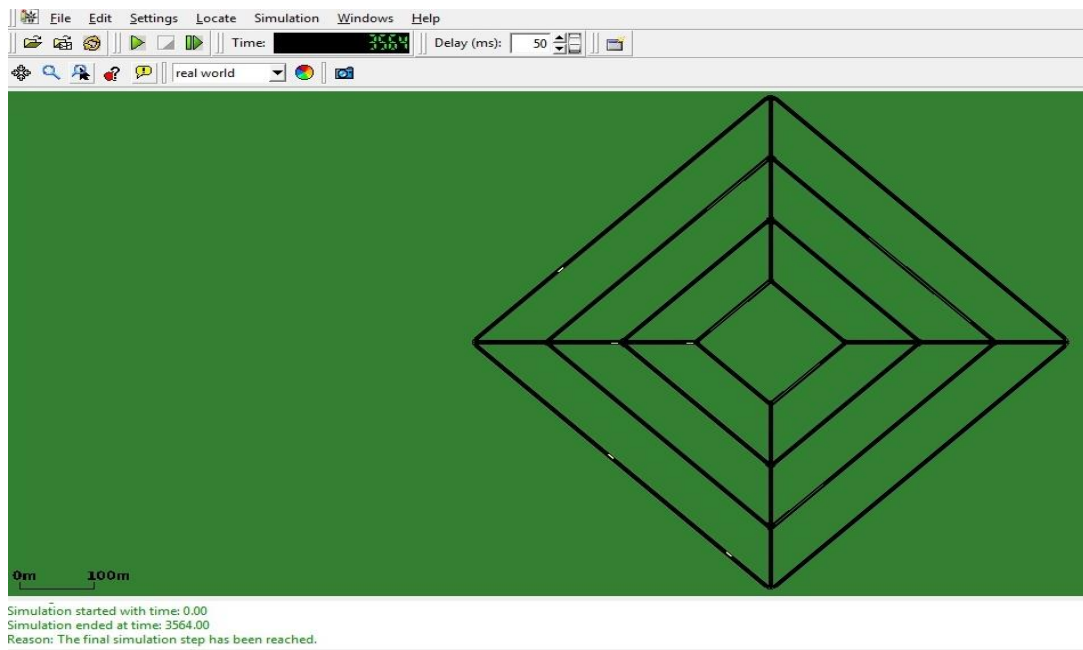
The data is created on the note pad and the option 'File' is selected from 'add turn flow wizard' to import the file. Once the turn flows are uploaded, **JTRROUTER** is then selected to uses flows and turning percentages at junctions as input.

Simulate:

After giving the demand to the traffic network, it is possible to run the simulation by choosing (*simulate>SUMO*) which would shift the network itself to the SUMO. The items in the window can be chosen for the kind of outputs which could be observed in the edge results.



Time for 'teleport' specify how long a vehicle may wait until being teleported, non-positive values disable teleporting and number for 'Seed' could be inserted to attain alternative outputs with the same simulation. The model is then simulated on the SUMO by choosing the large 'time delay' to slow down the simulation and a kind of visualization 'real world'. The representation of network on SUMO is shown below;



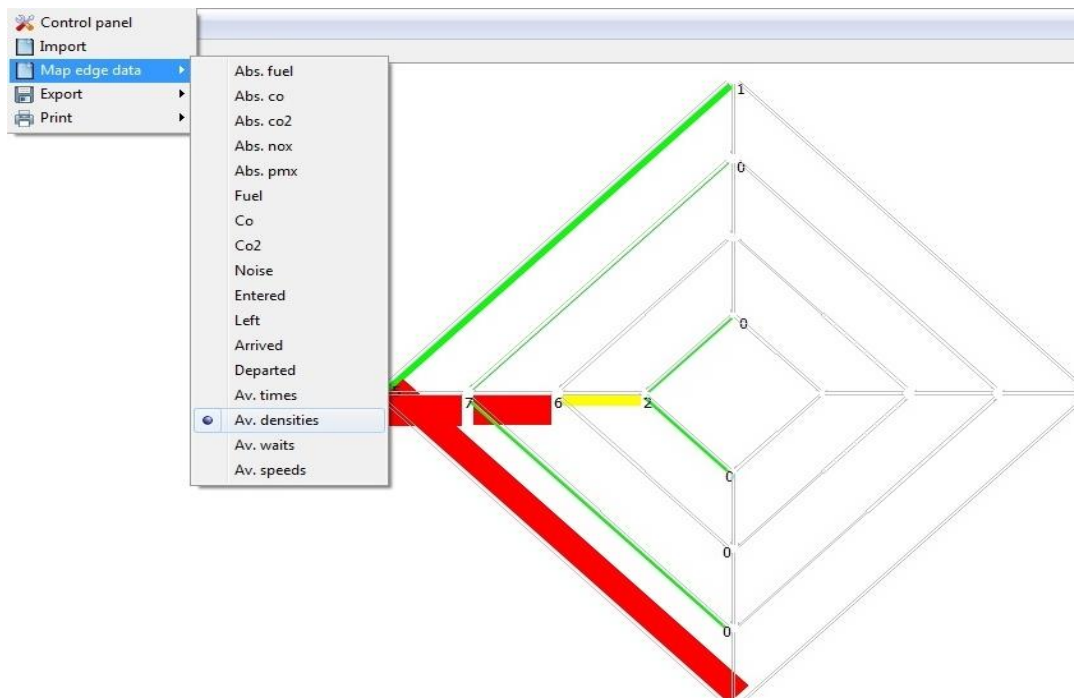
Results:

On the completion of the simulation, close the SUMO and choose (results>import) in SUMOPy to import the results from SUMO, which can be seen in the tables as below,

Results								
General	Edge Results	Trip Results						
		Abs. Fuel [ml]	Abs. CO [mg]	Abs. CO2 [mg]	Abs. NOx [mg]	Abs. PMx [mg]	Fuel [l/km/h]	CO [g/km/h]
	4/4to3/4	24890.164409	227895.473141	2430730.115550	698838.926213	39658.489461	47.023502	429.449309
	3/3to4/3	2595.753237	21407.482378	6510795.312428	73652.250730	3688.162347	6.400259	52.783693
	3/4to3/3	2594.827053	34812.776150	6508472.991481	69278.478173	6232.300408	29.550473	396.455713
	3/4to2/4	9479.317155	73670.033046	3776484.274122	269788.072833	12600.230576	18.241925	140.727683
	3/3to2/3	2355.142597	18800.355570	5907283.791611	66987.990506	3227.729913	5.806994	46.355390
	3/3to3/2	2121.165129	34063.664320	5320411.432008	54864.615093	6158.735899	26.513445	409.945891
	3/1to2/1	671.984961	6006.674202	1685505.724228	18962.457001	1043.912904	5.093882	45.532703
	3/2to3/1	1235.521345	13572.264034	3098995.719293	34119.865296	2400.723758	16.028396	171.117599
	3/1to4/1	765.615352	6395.977751	1920354.029274	21729.680626	1104.125559	5.803634	48.483761
	3/2to4/2	265.591207	2361.786787	666168.952175	7502.051353	410.396405	1.005456	8.941082
	3/2to2/2	263.487668	2257.661243	660892.736544	7466.517219	391.122087	0.997493	8.546891

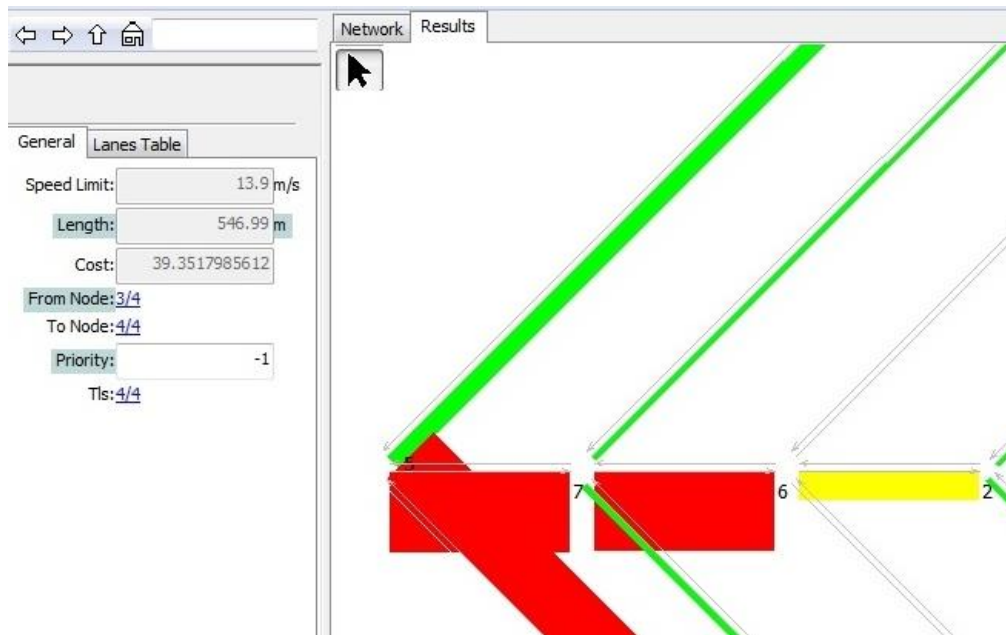
Results									
CO2 [g/km/h]	Noise [dB]	Entered	Left	Arrived	Departed	Av. times [s]	Av. Densities [veh/km]	Av. waits [s]	Av. speeds [m/s]
117946.652188	77.352500	0	97	0	99	116.532000	5.750000	9.616667	5.034083
16053.444072	29.350333	14	0	14	0	36.197000	0.652167	0.000000	2.678333
74119.952073	55.662500	55	55	0	0	44.510000	7.100333	12.433333	2.623333
45755.282894	62.166417	42	0	41	0	78.080667	1.662500	0.000000	6.606417
14565.386472	20.774333	13	0	13	0	20.093000	0.564333	0.000000	2.145333
66502.333790	43.164250	28	27	0	0	53.887583	6.987167	17.291667	1.483250
12776.726230	15.225833	10	0	10	0	6.615000	0.590500	0.000000	1.257000
40203.214667	29.971917	23	22	0	0	18.212917	2.677583	0.916667	1.941500
14556.958985	15.307000	12	0	12	0	6.068667	0.609167	0.000000	1.428500
2521.934326	4.438167	2	0	2	0	3.373000	0.116333	0.000000	0.372333
2501.960010	4.369833	2	0	2	0	4.223667	0.108667	0.000000	0.326500

It is also possible to observe the different outputs of the edge data by selecting any of the attributes by choosing (*Results>Map edge data*) in terms of graphical representation with normalized values such as;



In 'trip results', it is also possible to observe the durations, departure time and arrival time in seconds for the respective trip.

It is also possible to observe the speed limit, length and cost by selecting any of the edge as shown below;



Conclusion:

The study is presenting the implementation of microsimulation modeling using SUMOPy on SUMO simulator. SUMOPy user manual is described in the report for its application to manage large amount of data necessary to run complex multi-modal simulations. The objective of micro-simulation is, from the designer's point of view, to quantify the benefits of Intelligent Transportation Systems. Traffic is considered as essentially containing cars that move from one point to another weaving across lanes and creating queues on roads. Incidents and traffic calming measures may exist but motorbikes, bicycles, pedestrians, public transport, weather conditions and parking phenomena receive less attention. The interest in micro-simulation is to estimate traffic efficiency in terms of speed and travel time. Transport telematics or technological functions studied by most of the models are vehicle detectors, adaptive traffic signals, co-ordinated traffic signals, ramp metering and static and dynamic route guidance. Micro-simulation models provide a Graphical User Interface mainly to visualise simulation results. Model parameters can be user-defined and typical execution speeds are between 1 to 5 times faster than real-time. Most models use a time slicing approach in which computation is done at each time step. Identified limitations come essentially from an imperfect modelling of human behaviour and from the fact that the accurate modelling of a road network quite difficult. Micro-simulation is needed for short-term forecasts for on-line applications and for the evaluation and development of control strategies, large scale schemes and product performance tests. Most of the users consider it as a necessary or useful tool for traffic conditions analysis. Micro-simulation needs to be able to produce indicators to determine whether objectives that improve traffic efficiency are met.

References:

1. Towards the next-generation traffic simulation tools: a first evaluation, FEUP, University of Porto, Rua Dr. Roberto Frias.
2. **Amodel for traffic simulation** by Hans-Thomas Fritzsche, *Daimler-Benz AG*
3. Guidelines for Applying Traffic Microsimulation Modeling Software, California department of transportation.
4. Simulation Modelling Applied to Road Transport European Scheme Tests, <http://www.its.leeds.ac.uk/smartest>
5. Charypar, D., M. Balmer, and K. W. Axhausen (2008) A High-Performance Traffic Flow Microsimulation for Large Problems, Working paper, 509, Institute for transport planning and system (IVT), ETH Zurich, Zurich.
6. A Traffic Simulation Modeling Framework for Rural Highways by Andreas Tapani, Department of Science and Technology Linköpings Universitet.
7. 'Sustainable transportation engineering', Professor Joerg Schweizer, DICAM, University of Bologna.
8. SUMO (Simulation of Urban Mobility), <http://sumo-sim.org/userdoc/Tutorials.html>