# Documentation

## WHAT IS IT?

This is a model which simulates car and bus/tram traffic in Augsburg, specifically between the districts Stadtbergen, Göggingen and the Innenstadt. People either use their cars or public transport to travel to one of their random destinations (Stadtbergen or Göggingen), performing some activity and then returning to their home. Both the actual travel time as well as their happiness upon arriving are stored and have an impact on individuals on whether they would consider changing their mode of transport or not.

## HOW IT WORKS

### World Generation

Using the gis extension, this model loads in a simplified road network with the roads being links and the nodes being a stationary breed. Both inherit some data such as their calculated length in ArcGis Pro, their maximum allowed speed limit from OSM and some data about the node being a bus/tramstop or not. Both the tram and hypothetical busroute are highlighted with thicker and colored links.

### Agent Placement

People can either move to their destination by car or using public transport, the ratio is determined from the modal split in the city of Augsburg. Car users can be placed anywhere except certain nodes such as the B17 highway, while public transport users will only be placed on bus/tramstops. The trams and buses will always start from their initial location (Königsplatz, Stadtbergen or Göggingen). Additionally, another breed called "filler-vehicles" will be placed on either the north/south end of the B17 highway, the Königsplatz, Stadtbergen and Göggingen as well as some randomly placed cars. Their purpose is to fill up the world with additional traffic especially on the B17 highway and the two main streets towards the Königsplatz.

### Pathfinding

Using the nw extension, a Djikstra-algorithm is performed to find the shortest path from the starting point of people and trams/buses to their destinations. While the people generate at random nodes each simulation and have a random destination, the buses/trams stay the same for every simulation (move from initial stop to last stop and then back). The path is stored as a list containing the nodes an agent has to pass to arrive at its destination. If an agent uses public transport and has to transfer to another bus/tramline, then this agent will have two lists (one until the transfer and another one for the whole route).

### People behaviour

Once the simulation starts, both car users and public transport users will move towards their destinations along the links.

### Cars

Car users will drive along the links according to the maximum allowed speed limit. If a lot of cars are infront of the car agent, then it will slow down a bit and cause a bit of their happiness to decrease. The list containing the nodes will get shorter and shorter depending on how much is left until the final destination. Once one car agent reaches their destination, they will stop moving and get some additional time added to include finding a parking lot (this is randomly picked). After that, they will stay at their destination performing some sort of activity such as working which is a counter that will decrease every tick. Upon reaching 0, the agent will first calculate its original list back, swap the current location with its destination and then reverse the node list (this is essentially the way back).

### Public transport Users

People using public transport will be generated on a random node (except destination nodes and transfer nodes) which is also a tram or busstop. A random amount of time will be generated for every public transport agent to include an approximation of the time they need to walk from their home to the bus or tramstop (a few minutes). They will wait until a bus or tram enters their short radius and check if either their destination or transfer-destination matches the last node of the tram/bus (reminder: every moving agent has its list of nodes that it follows to reach its destination). If they don't match, the public transport agent will wait on the stop. However, if they do match, they will link together using "tie" which is provided by Netlogo. Once public agents reach their destination or transfer-destination, they will untie with the bus or tram they were traveling on. If they reached their final destination, they will first add some time to include their walk from the stop to their actual destination (a few minutes usually) and then begin to perform their activity. However, if they need to transfer to another tramline or busline (e.g. leaving a tram in Line 1 at Königsplatz to enter another tram in Line 3 to travel towards Stadtbergen) they will first move towards the next stop, wait there and then link with the tram or bus that meets criteria of having the same last node in their list as the destination node of the public transport agent. The procedure for getting home after having performed an activity is similar to the one the car agents use (recalculating the old list back, then reverse it). However, one difference is the fact that the destination is a random node (their original starting point) now and not the ending node of a bus or tramline. Therefore, the public transport agent checks if their destination is somewhere in the list of the bus or tram that it needs to take. The agent will tie once again if this is the case and untie once they are near the destination (home) again.

### Tram/Bus/Filler-vehicles behaviour

Both trams and buses belong to one line and have their own list of nodes that they need to travel through to reach their final destination. In this simplified model, the destinations are located in Göggingen, Stadtbergen and the Königsplatz.

### Trams

There are 4 trams in total with 2 on each line. Those will travel from their starting point to their endstation while stopping for a short amount of time, when they are on a tramstop along the way (marked as slightly larger green points).

**Buses**

If the hypothetical line (called line 500 in this example) is generated as well, then two buses will travel similarly to the trams (back and forth).

**Filler-vehicles**

Similarly to car agents, this group starts at a location and is assigned a destination. They serve as a means to fill up the overall traffic in this model with a focus on the two main streets towards the Königsplatz as well as the B17 highway, replicating what traffic count data suggests for atleast the B17 and the Gögginger Straße. Once those vehicles reach their goal, they will be moved back to their inital location and drive towards their destination once again.

**Output**

Every time a car or public transport agent reaches its destination, it will store some information in a stationary reporter. This includes:

- the ID of the current agent

- how many minutes it took in total (includes time spent waiting, traveling and going to/from the bus/tram stop or finding a parking lot)

- what mode of transport the agent is using

- time spent waiting (only important for public transport agents)

- time spent traveling (units in minutes for public transport agents and hours for car agents)

- the happiness upon arriving at the desired destination

A procedure writing the values of the attributes of the reporter starts at 30000 ticks (right at the end) in a .csv file.

**Note:** If one wishes to run the model multiple times, the .csv files need to be moved to a different directory since this model would overwrite an already existing .csv.

## HOW TO USE IT

The sequence of the buttons and sliders is shown in the Interface tab with numbers inside brackets "()":

**(1)** Choose the scenario with the switches down below

- new-route?

  **on:** includes the hypothetical express bus route between Stadtbergen and Göggingen

  **off:** doesn't include the express bus route (status quo)

- load-labels?

  **on:** load a few labels (for better readability/orientation)

  **off:** don't load in any labels

**(2)** Create the area of interest

**setup:** generates the Netlogo world by loading in the needed geodata as well as including the new express bus route if it has been chosen in (1)

**(3)** Select sample size of car and public transport agents, then place them randomly

**number-of-people** specifies the amount of car and public transport agents (in total), the ratio equates to the modal split of the city of Augsburg

**place people** places the people on random nodes (with some exceptions such as the B17 highway)

**(4)** Calculate the shortest path using Dijkstra

**calculate paths** calculates the shortest distance using the Dijkstra-algorithm included in the nw-extension and stores the list of nodes that need to be passed until the destination

**(5)** Select time a bus or tram stops at a stop

**wait-time** specifies the amount of ticks (seconds) a mode of public transport will spend at a stop

**(6)** Select happiness threshold and the impact of sources which decrease happiness over the course of the simulation

**happiness-threshold** specifies a threshold (a number between 1 and 100) in which an agent will consider switching its mode of transport if its happiness is lower than the threshold after a journey

**transfer-loss** specifies the reduction in happiness for when public transport agents have to transfer from one tram or bus to another one to reach their destination (same units as happiness-threshold)

**congestion-loss** specifies the reduction in happiness for when car agents have to slow down due to other cars or buses/trams being ahead of them (same units as happiness-threshold)

**waiting-loss** specifies the reduction in happiness for when public transport agents have to wait for a bus/tram to arrive (same units as happiness-threshold)

**parking-last-loss** specifies the reduction in happiness for when both car and public transport agents have to either find a parking lot (car only) or travel from the final bus/tramstop to their destination (public transport only) (same units as happiness-threshold)

**(7)** Start the simulation

**go:** Simulation will start and run once this button is pressed (currently, a piece of code will make it stop once a bit over 8 hours have passed; equivalent to 30000 ticks)

## THINGS TO NOTICE

This simulation focuses more on the final output e.g. the overall travel time, the overall happiness and the percentage of people who are considering to switch their mode of transport. However, if one chooses to observe the simulation, then there are some things to notice:

- the overall traffic flow of the study area as well as which roads are used more often and which aren't

- the monitors as well as the plot to get an idea of the state (average happiness of the people after their trips, how many are willing to change from car to public transport and vice versa)

- (not used in the experimental setup) the number next to buses/trams since this indicates the amount of people that are currently travelling (this could be helpful for extensions to evaluate the capacity of the new express bus line for example)

## THINGS TO TRY

The more interesting slides are the ones in **(5)** and **(6)** since they affect both the travel time and the happiness. With accurate empirical data, those could be adjusted to better represent the happiness of the agents.

## EXTENDING THE MODEL

This simulation is very simple baseline of representing the traffic flow and happiness of the agents in this particular area of interest. This model can be refined and extended in the following ways:

- use a **more accurate traffic network** which includes more roads, traffic lights on important crossovers, the introduction of one-way streets (e.g. the tramline 3 splits up in Pfersee due to two one-way streets) and the introduction of remaining tramlines and already existing buslines (however, this will increase the computation time for both the model and the pathfinding algorithm)

- introduce **more participants** concerning the **mode of transport** such as motorcycles, pedestrians, bikes, e-scooters and buses

- additionally, add agents without a destination into this model; since traffic isn't just a system that is completely limited to the study area, it would be more realistic to have cars travel in and out of the model (e.g. people that drive through the B17)

- apply more accurate **empirical data** concerning the actual modal split in this particular region, the variety in cars (different types of fuel, possibly e-cars), socio-economic (estimate on salaries, how many people actually do own a car, which age groups are present since for example a dominance of old age people might lead to far less cars being used and more public transport; age is also important to understand what kind of school/university and workplace traffic flow this region can expect) and psychological (happiness of the current status quo concerning how fast and comfortable people can reach their destination in this area of interest, happiness concerning the pricing, happiness concerning the comfort on both bus/tramstops and inside buses/trams)

- more room for **interaction between the agents** such as cars stopping at traffic lights leading to a queue of cars, stopping to let pedestrians pass or cars waiting for buses or trams in one-lane streets

## NETLOGO FEATURES

### Netlogo extensions:

- nw

- gis

- csv

Due to how links work as a datatype in Netlogo, a solution was needed to make the agents move in realistic steps in between the links (move-to would just make them jump from one end to another), which are essentially the entire world they can travel in. Using the **nw** extension proved to be very helpful not only for the calculation of the shortest path but also as a possibility to provide the agents with a list of sequential nodes.

## RELATED MODELS

**Related interest:**

[1] Simulation of a small town with interactions between pedestrians, traffic lights and vehicles by Jiri Lukas

**Netlogo Community Hub Link:**

- http://ccl.northwestern.edu/netlogo/models/community/Town%20-%20Traffic%20&%20Crowd%20simulation

[2] Project Autonomous Bus in the

"Innovationspark Augsburg" by Katharina Fogelberg, Sandra Kempter, Johannes Krämer, Tanja Pfeffer and Julia Subal in 2018

## CREDITS AND REFERENCES

This model was created at the Institute of Geography, University of Augsburg.

**Other references:**

[1] Netlogo NW Extension for Network Analysis

**Github Link with information about computation time etc.:**

- https://github.com/NetLogo/NW-Extension

**Netlogo Manual Link:**

- https://ccl.northwestern.edu/netlogo/docs/nw.html

[2] Dijkstra by Matthias Benedek in 2014 (Model created at the Institute of Geography, Augsburg)

- generating the links and nodes from geodata used as a basis from this model