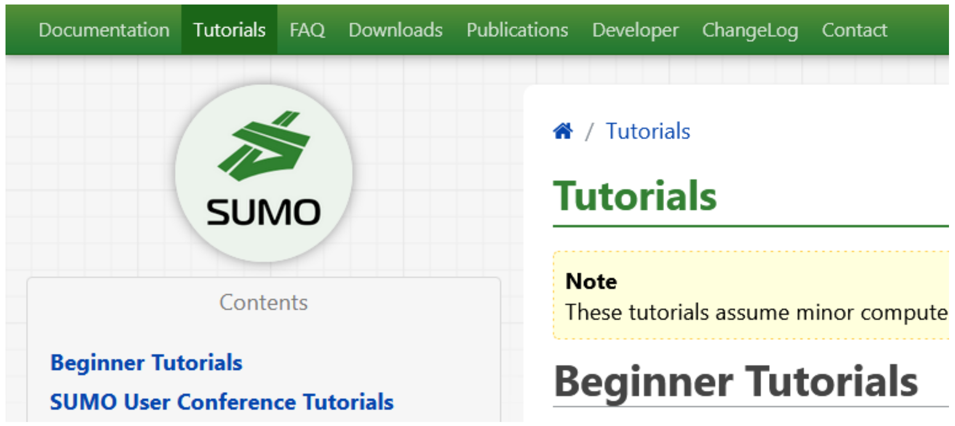
SUMO Tutorial

Jakob Erdmann

SUMO2023, Berlin

Outline

- 3-Click network generation with netgenerate
  - comparing networks with netdiff
- 3-Click scenario generation with osmWebWizard.py
- Simulating bicycles
  - Preparing the network
  - Defining traffic
  - Analyzing and plotting results
  - Running Scenarios repeatedly



Tutorials

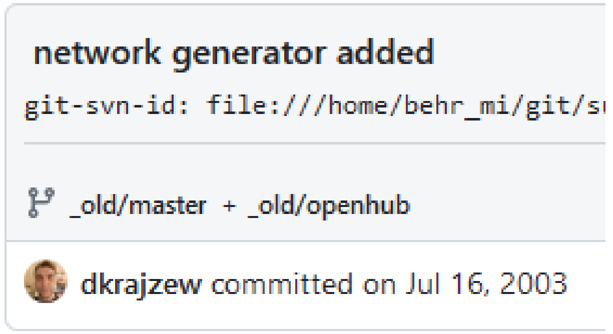
Note
These tutorials assume minor compute

Beginner Tutorials

- **Prerequisites**
  - SUMO 1.17.0
  - Python: python.org/download/
  - Data files: sumo.dlr.de/daily/sumo2023_tutorial.zip

Netgenerate

- exists as command line tool for almost 20 years
  - more details in the SUMO2019 tutorial
- now with UI!

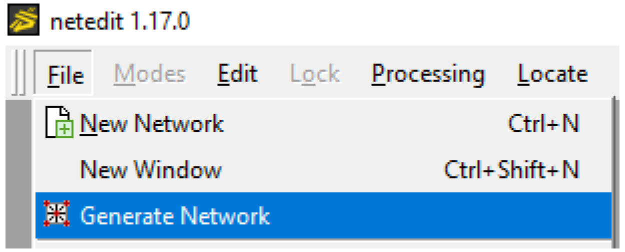


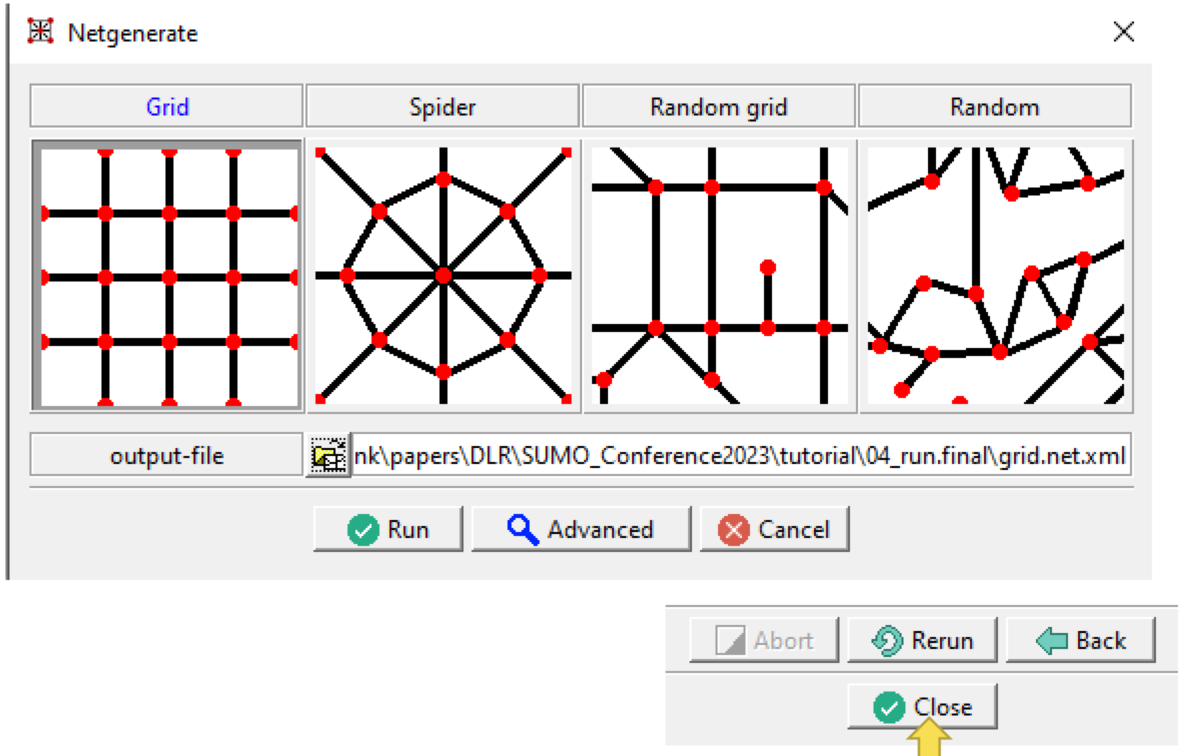
network generator added

git-svn-id: file:///home/behr_mi/git/s

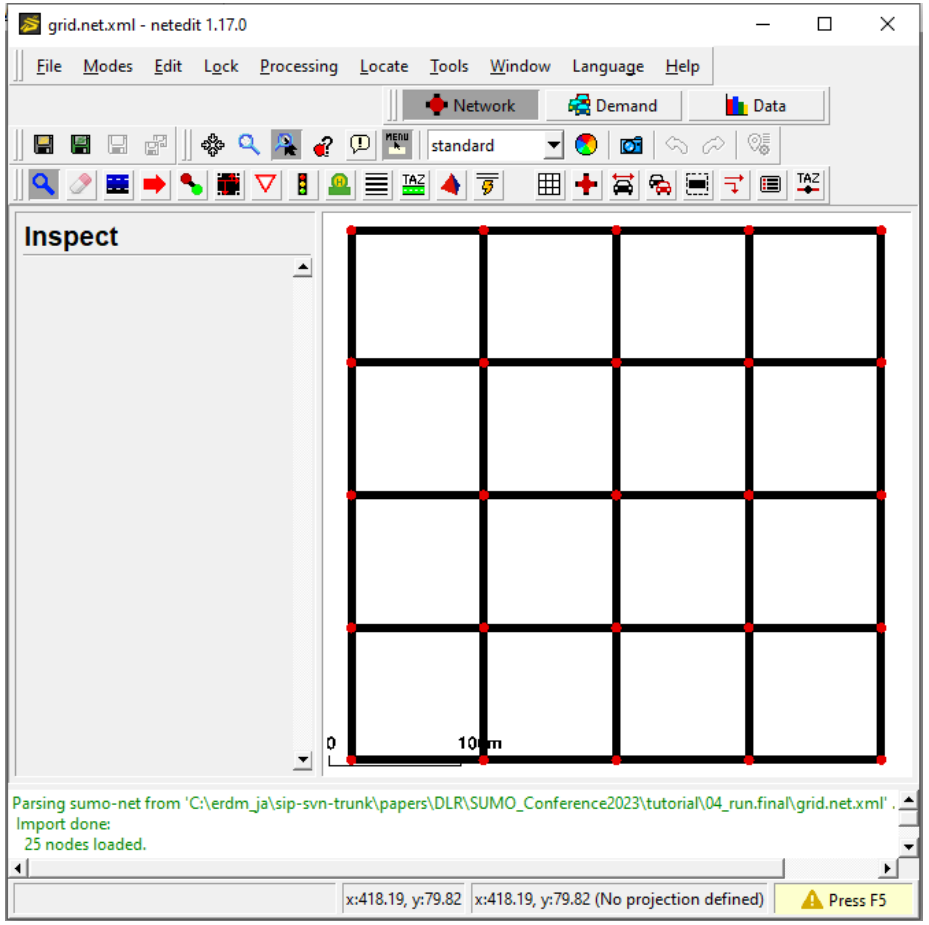
_old/master + _old/openhub

dkrajzew committed on Jul 16, 2003





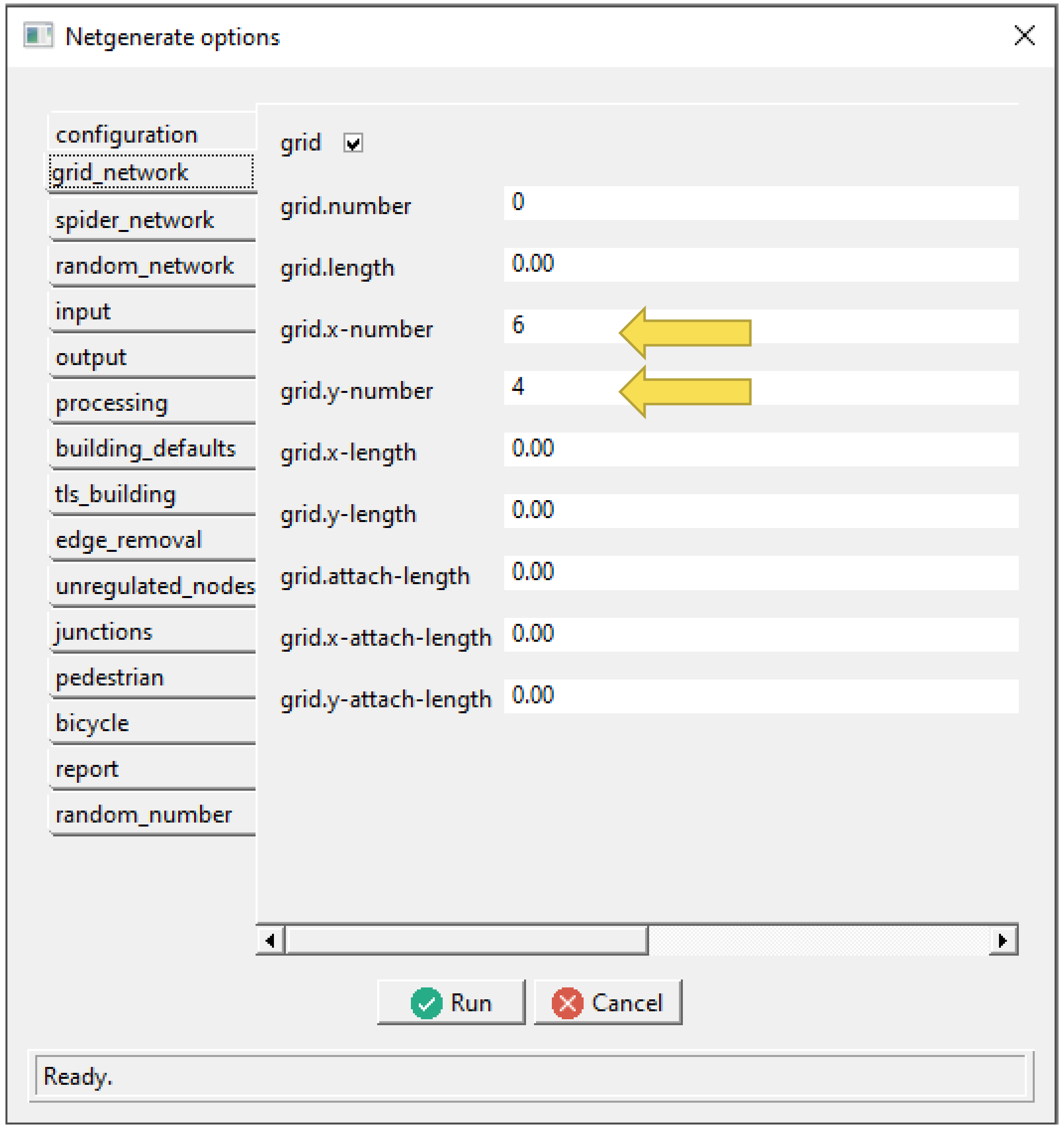
nk\papers\DLR\SUMO_Conference2023\tutorial\04_run.final\grid.net.xml



```
Parsing sumo-net from 'C:\erdm_ja\sip-svn-trunk\papers\DLR\SUMO_Conference2023\tutorial\04_run.final\grid.net.xml' .
Import done:
25 nodes loaded.
```

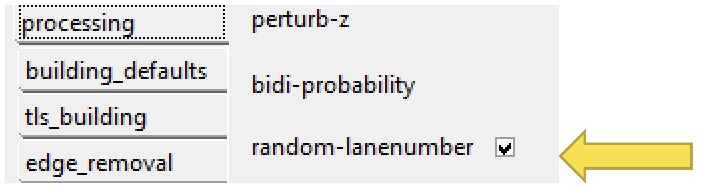
x:418.19, y:79.82 x:418.19, y:79.82 (No projection defined) Press F5

Netgenerate - Advanced

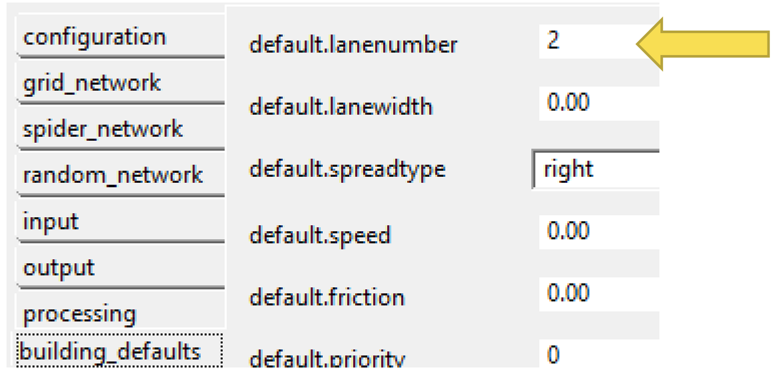


Netgenerate options

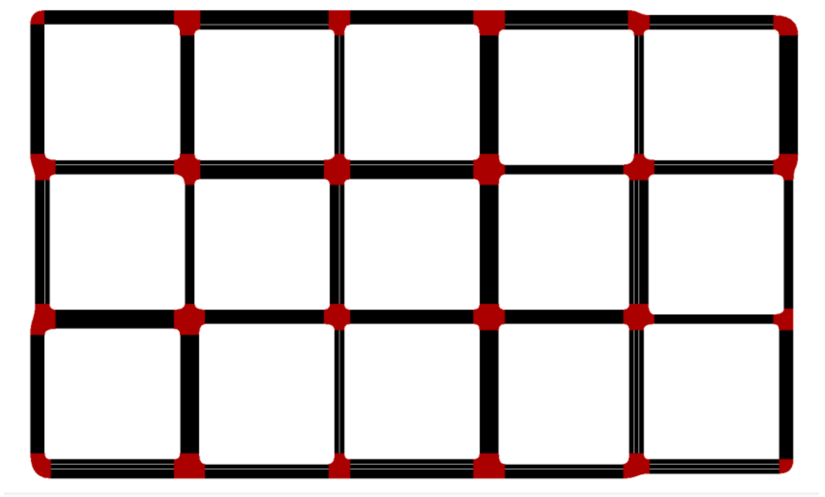
grid	✓
grid.number	0
grid.length	0.00
grid.x-number	6
grid.y-number	4
grid.x-length	0.00
grid.y-length	0.00
grid.attach-length	0.00
grid.x-attach-length	0.00
grid.y-attach-length	0.00



perturb-z	
bidi-probability	
random-lanenumber	✓



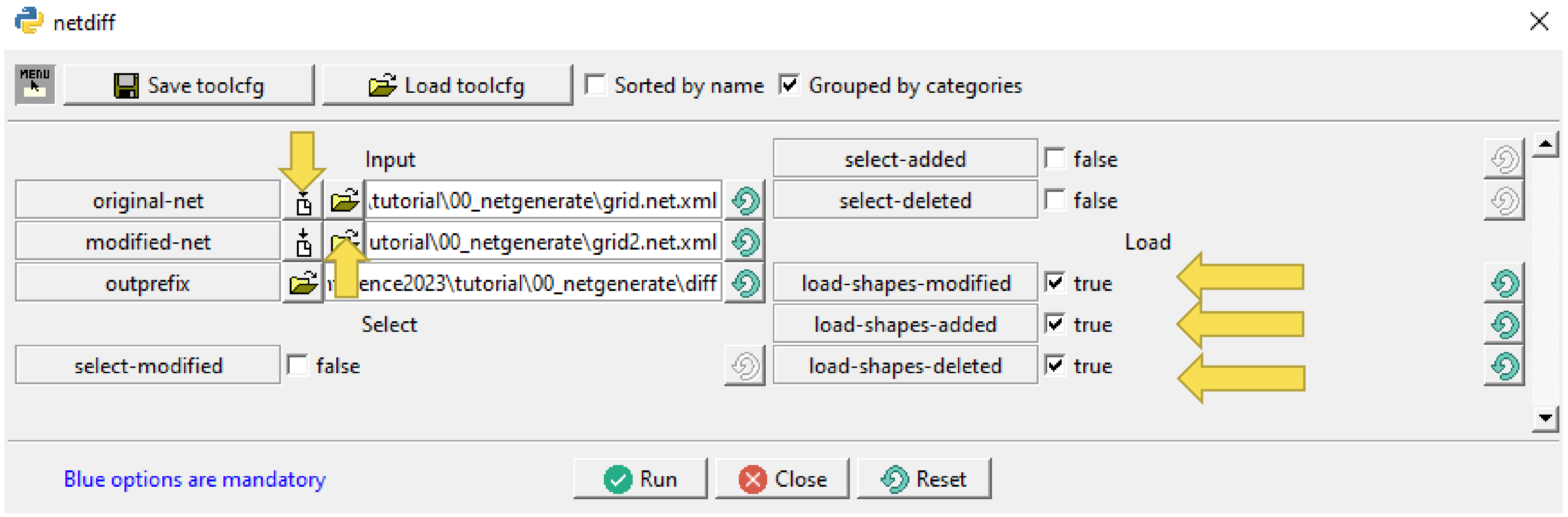
default.lanenumber	2
default.lanewidth	0.00
default.spreadtype	right
default.speed	0.00
default.friction	0.00
default.priority	0



grid2.net.xml

Visual network difference

- netdiff.py: command line tool for computing differences in networks
  - B.net.xml - A.net.xml = diff.xml (**netdiff**)
  - A + diff.xml = B.net.xml (**netconvert**)
- see the differences in netedit
  - open grid.net.xml


netdiff

Save toolcfg | Load toolcfg | Sorted by name [ ] | Grouped by categories [✓]

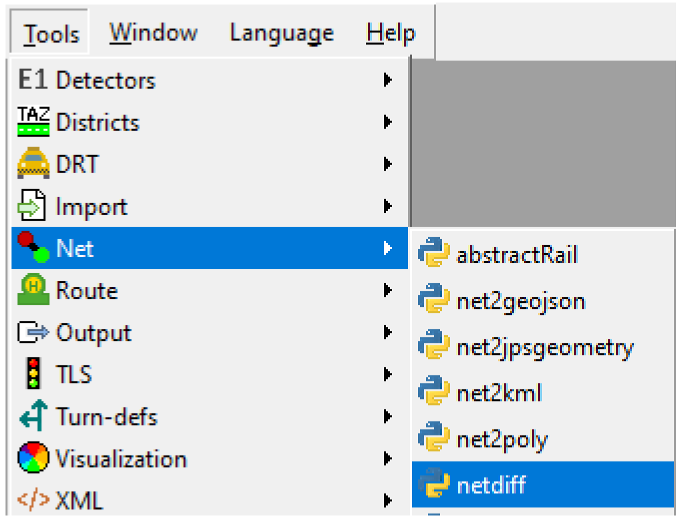
Input
- original-net: tutorial\00_netgenerate\grid.net.xml
- modified-net: utorial\00_netgenerate\grid2.net.xml
- outprefix: ence2023\tutorial\00_netgenerate\diff

Select
- select-modified: false
- select-added: false
- select-deleted: false

Load
- load-shapes-modified: true
- load-shapes-added: true
- load-shapes-deleted: true

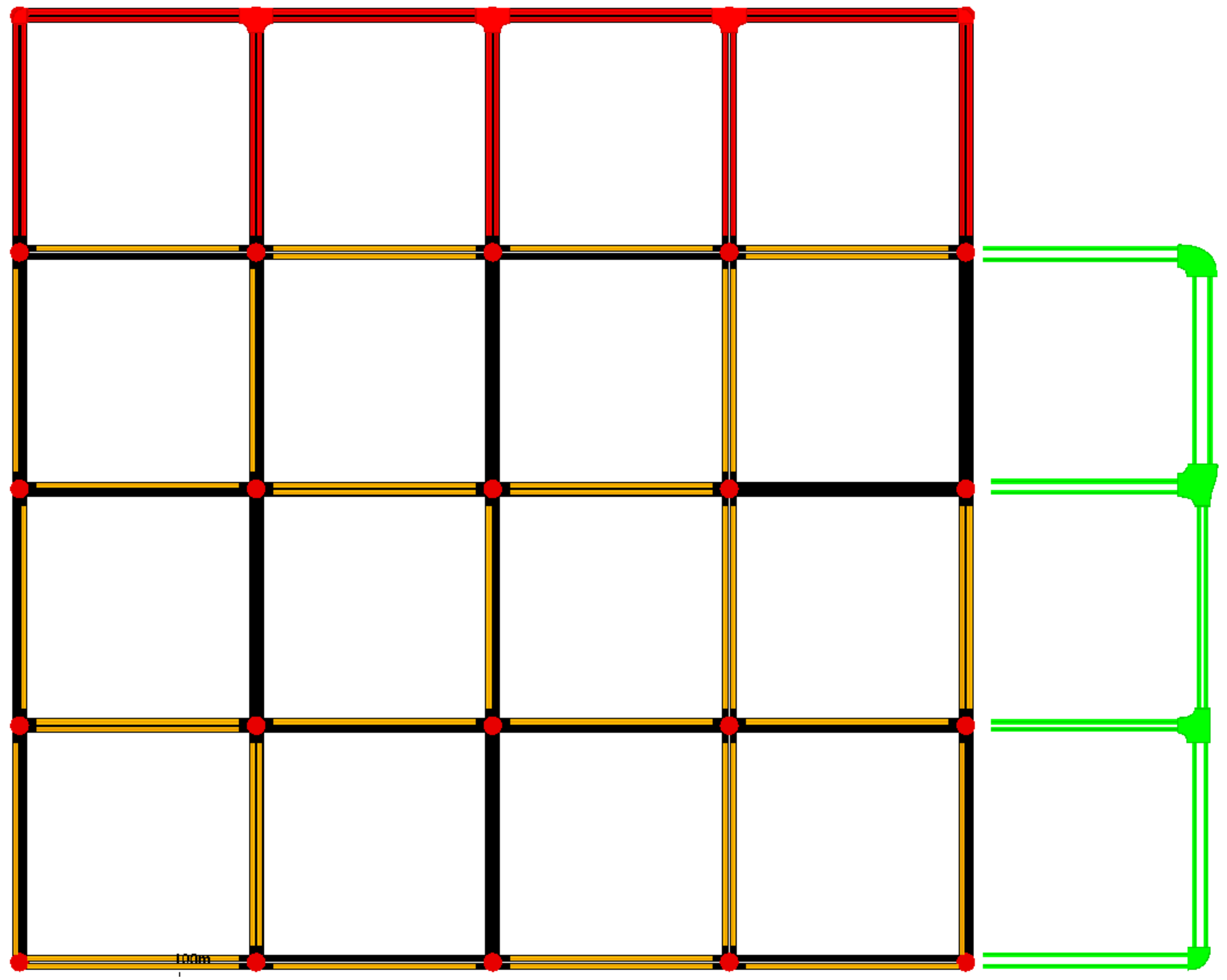
Blue options are mandatory

Run | Close | Reset



folder: 00_netgenerate

Visual network difference



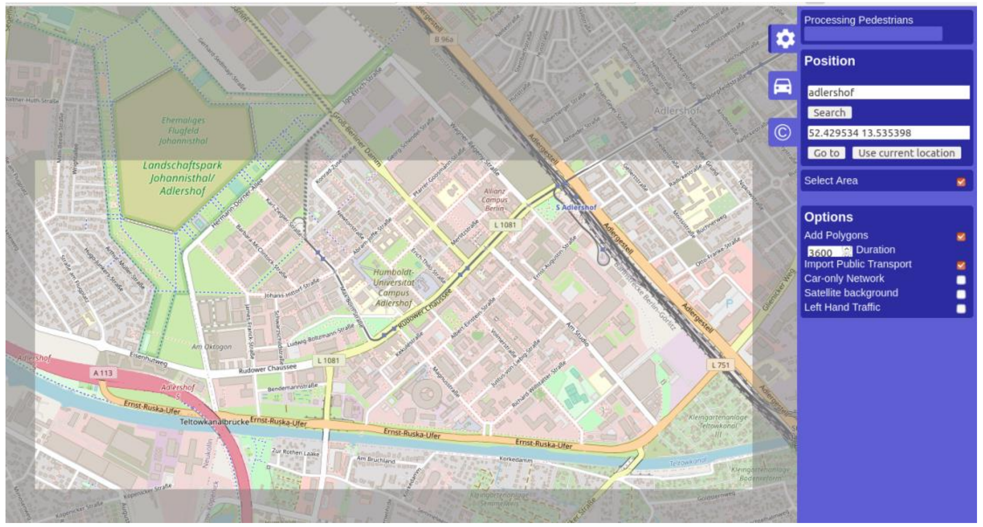
Changes represented as polygons

- created elements
- modified elements
- deleted elements

folder: 00_netgenerate

osmWebWizard

- tools/osmWebWizard.py
- OpenStreetMap network data
- **Random traffic**
- Configure
  - Area
  - road types
  - Traffic modes
  - Traffic volume
  - Fraction of through-traffic
  - Public Transport
  - Scenario duration
  - Building Shapes and Points-of-Interest (cosmetic)
  - Satellite background (cosmetic)
- Generated files allow rebuilding and adapting the scenario
- Example data in `01_wizard`


Processing Pedestrians

Position

adlershof

52.429534 13.535398

Select Area ✓

Options

Add Polygons ✓

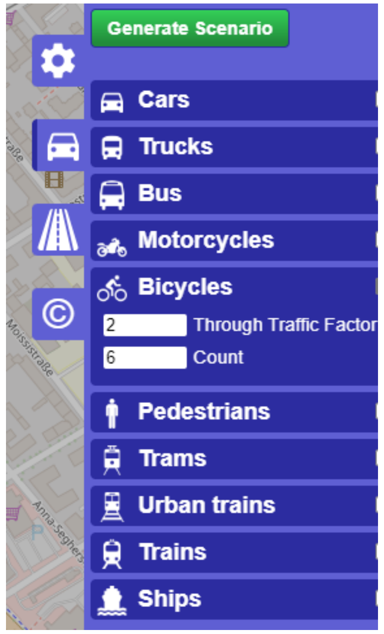
3600 Duration

Import Public Transport ✓

Car-only Network ☐

Satellite background ☐

Left Hand Traffic ☐



Bicycles: 2 Through Traffic Factor; 6 Count

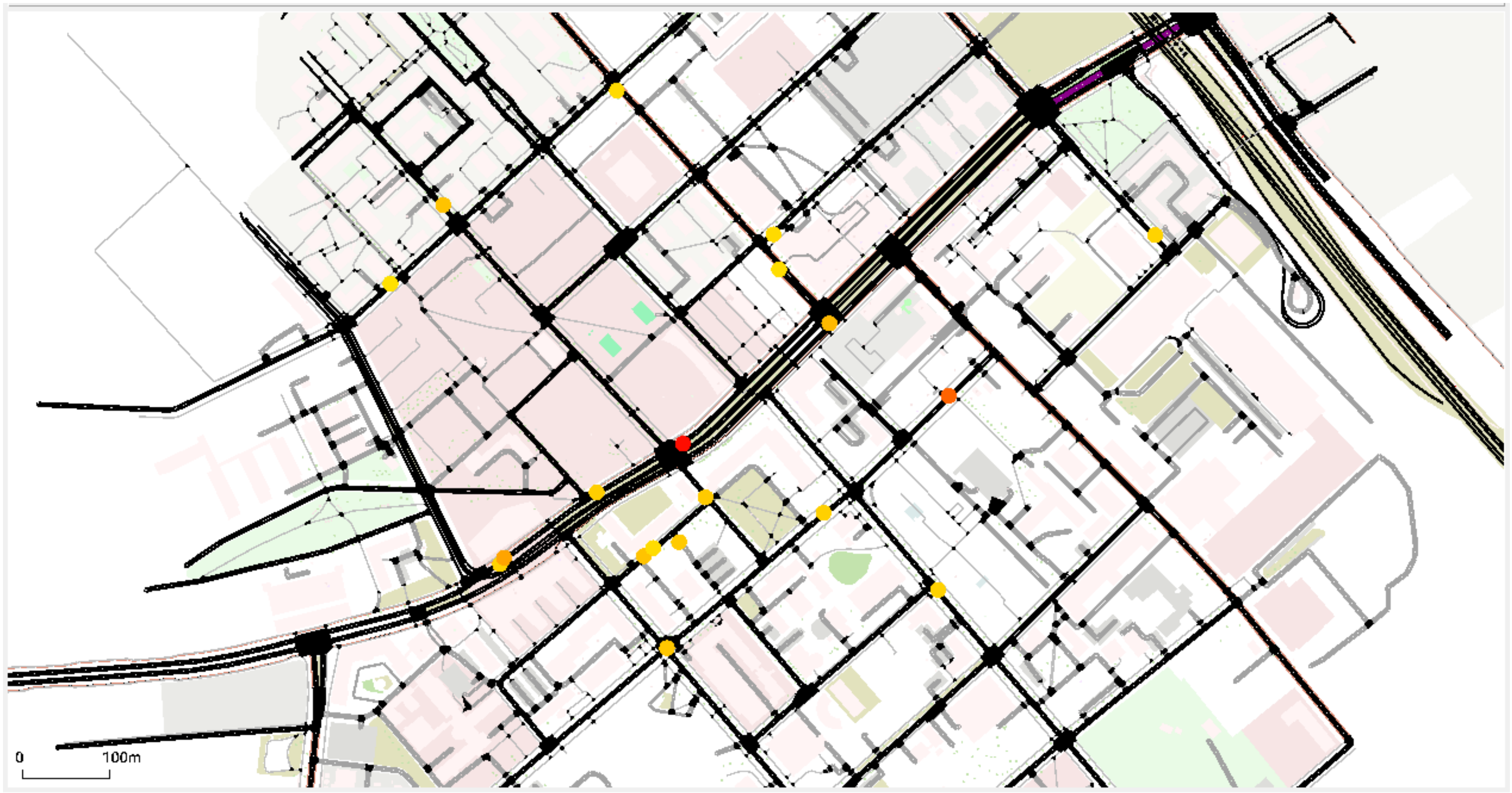
osmWebWizard - Generated Files

- Scenario input
  - `osm.sumocfg`: configuration file (load with **sumo**, **sumo-gui**)
  - `osm.net.xml.gz`: simulation network
  - `osm.bicycle.trips.xml`: bikes (we didn't generate cars this time)
  - `osm.poly.xml.gz`: building shapes and POIs
  - `osm.view.xml`: sumo-gui settings for delay, colors,...

- Rebuilding:
  - `osm_bbox.osm.xml.gz`: raw OSM data
  - `osm.netccfg`: rebuild network and stops (**netconvert**)
  - `osm.polycfg`: rebuild shapes (**polyconvert**)
  - `build.bat`: rebuilt traffic (cars, persons, public transport schedule,…)

folder: 01_wizard

osmWebWizard - Simulation



osmWebWizard - Simulation

- Traffic is random and only contains bicycles
    - activating bicycle demand sets network building options for cycling infrastructure!

- We have warnings for 3 traffic light controlled intersections:

```
Warning: At actuated tlLogic 'cluster_...', linkIndex 8 has no controlling detector.
```

-
    - indicates that detector-based traffic actuation is not working for some approaches due to connection and phase layout.
    - can be fixed with either
        - global option **--tls.actuated.jam-threshold** (making all actuated tls smarter)
        - traffic light `<param key="jam-threshold value" value="30"/>`
        - setting the traffic light type to `"static"`

Next Goal - Detailed Bicycle simulation

- bicycles should overtake each other on a bicycle lane
  - widen the bicycle lanes
  - activate **sublane model** so they can overtake on a single lane
  - add more bicycles so they actually meet each other on the road
  - ~~configure the spread of desired speeds so they ***want*** to overtake~~

works out of the box in 1.17

- then we can compare different scenarios and make plots!

Network Editing - Widen the bicycle lanes

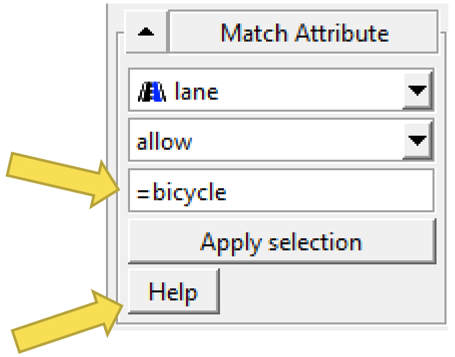
- **Option 1:** Rebuild from OSM with different typemap file
  - {SUMO_HOME}/data/typemap/osmNetconvert.typ.xml
  - {SUMO_HOME}/data/typemap/osmNetconvertBicycle.typ.xml
  - change values directly or modify a copy and adapt osm.netcfg

```
<type id="highway.cycleway"  numLanes="1" speed="5.56"  priority="1"
                             oneway="false" width="2" allow="bicycle"/>
 <type id="cycleway.lane"                 bikeLaneWidth="2.0" allow="bicycle"/>
 <type id="cycleway.opposite_lane"        bikeLaneWidth="2.0" allow="bicycle"/>
 <type id="cycleway.track"                bikeLaneWidth="2" allow="bicycle"/>
 <type id="cycleway.opposite_track" bikeLaneWidth="2" allow="bicycle"/>
```

- netconvert -c osm.netccfg

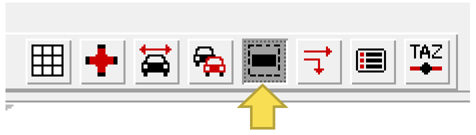
Network Editing - Widen the bicycle lanes

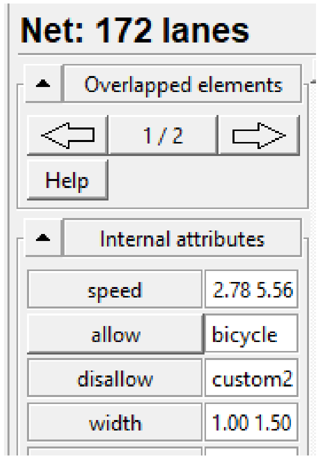
- **Option 2:** Use Netedit
- select mode (S)
- select lanes where attribute `allow` is `=bicycle`
  - the '=' triggers an exact match (see 'Help')
- inspect mode (I)
- **Shift-click** on any of the selected (blue) lanes to inspect them all at once
  - alternative: toggle clicks to target lanes (Alt+5)
  - by default clicks target edges
- set width to 2
- recommended:
  - select mode (S)
  - clear selection (ESC)



Match Attribute

lane	
allow	
=bicycle	
Apply selection	
Help	





Net: 172 lanes

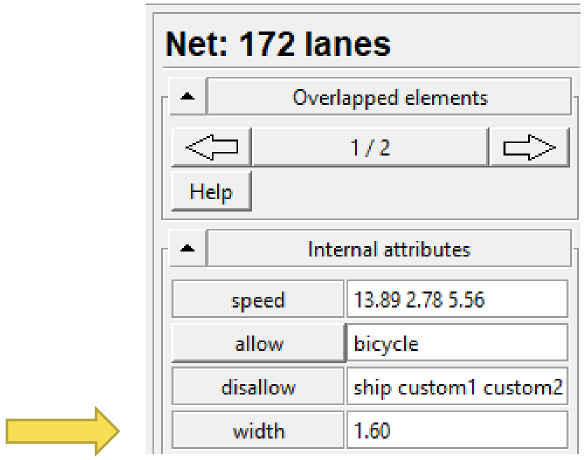
Overlapped elements

1 / 2

Help

Internal attributes

speed	2.78 5.56
allow	bicycle
disallow	custom2
width	1.00 1.50



Net: 172 lanes

Overlapped elements

1 / 2

Help

Internal attributes

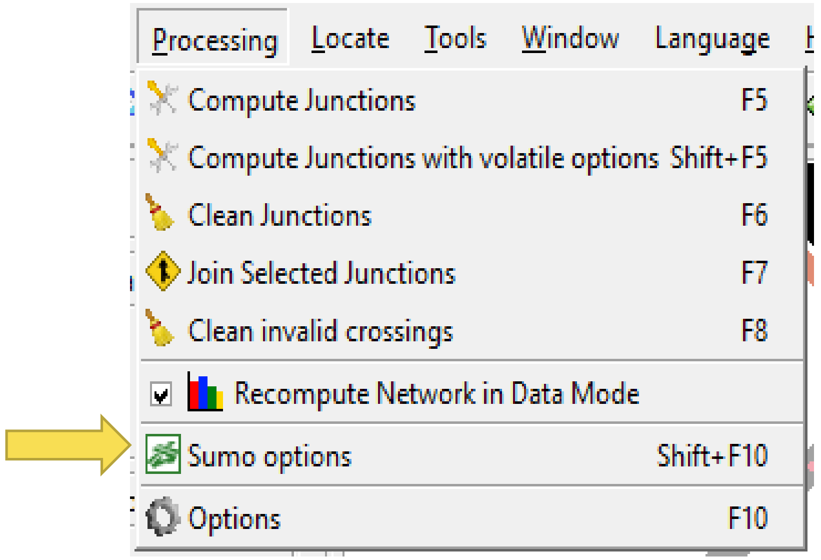
speed	13.89 2.78 5.56
allow	bicycle
disallow	ship custom1 custom2
width	1.60

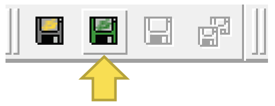
folder: 02_netedit

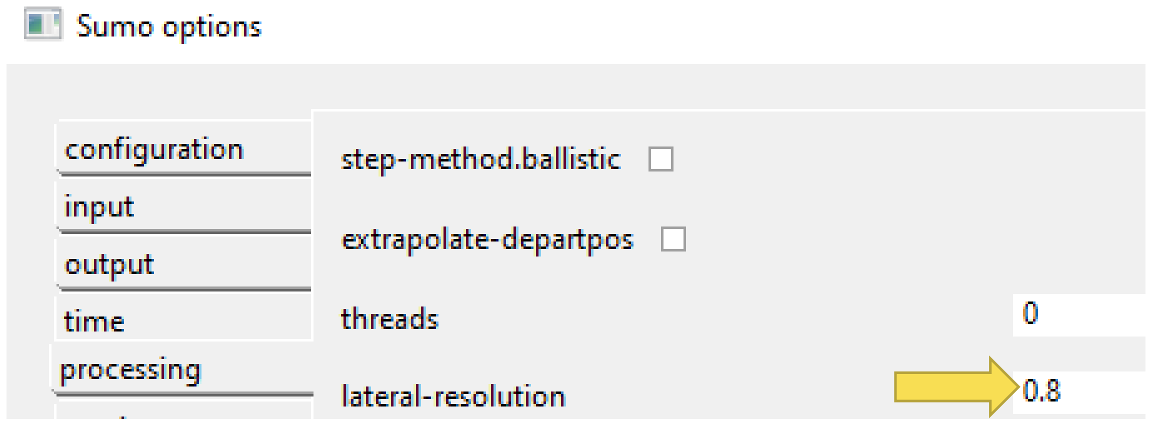
Edit .sumocfg

- Activate the **sublane model**
  - Sumo option dialog (Shift-F10)
  - processing
    - lateral-resolution: 0.8
    - *Bonus* jam-threshold: 30
  - OK

  - save .sumocfg


Processing
Locate
Tools
Window
Language
Compute Junctions F5
Compute Junctions with volatile options Shift+F5
Clean Junctions F6
Join Selected Junctions F7
Clean invalid crossings F8
Recompute Network in Data Mode
Sumo options Shift+F10
Options F10



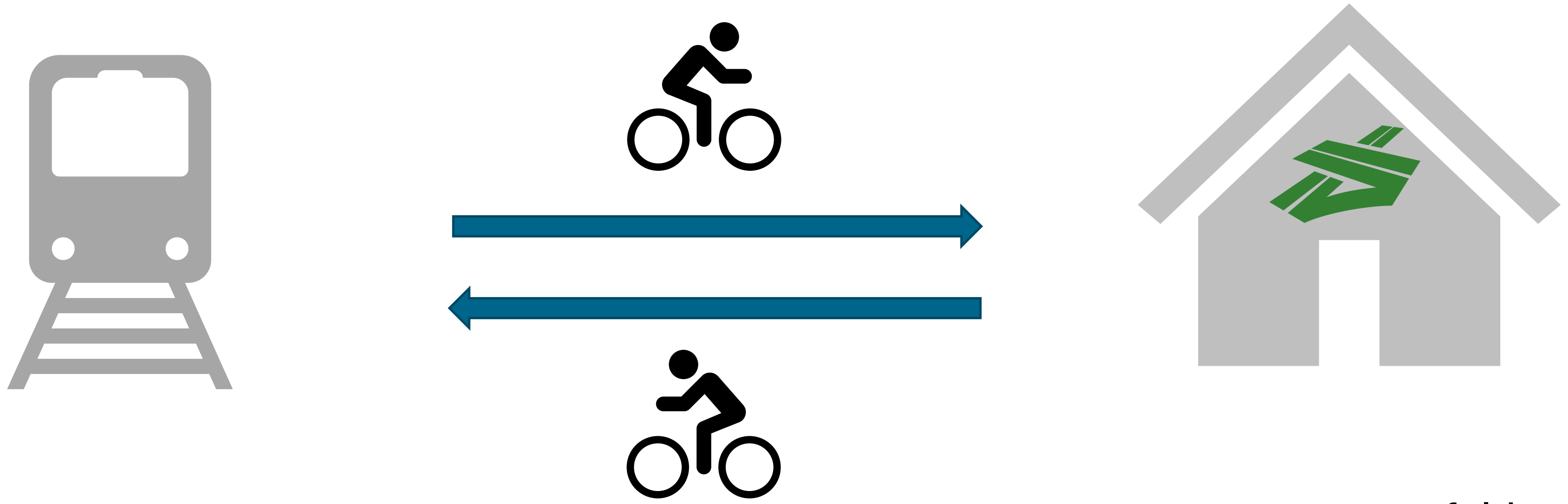


step-method.ballistic ☐
extrapolate-departpos ☐
threads 0
lateral-resolution 0.8

folder: 02_netedit

Define bicycle flow

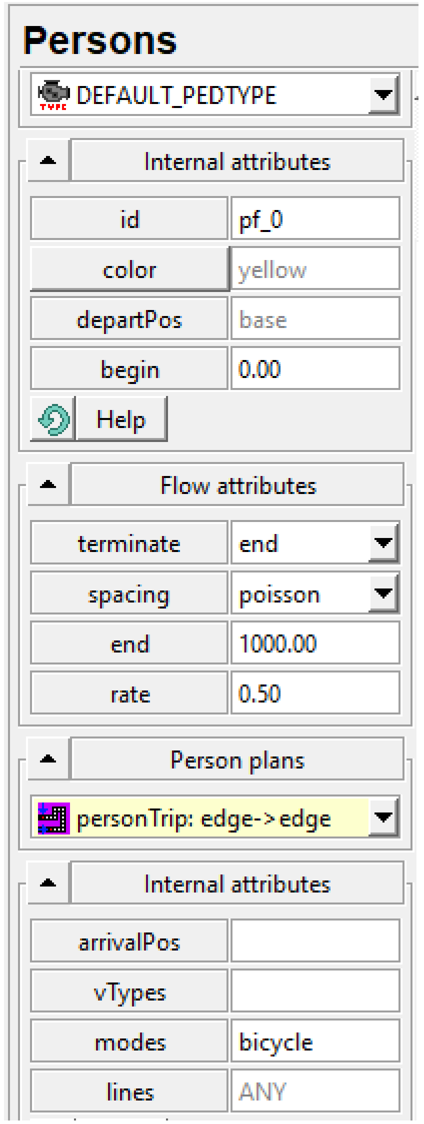
- simple vehicle flows were explained in the 2022 tutorial
- define a personFlow with bicycles instead!
  - ride to the DLR by bike
  - stay for the conference
  - cycle back to the train station



folder: 03_bicycles

Define bicycle flow (2)

- demand supermode (F3)
- person mode (P)
  - personFlow, spacing=poisson, end=1000
  - personTrip: edge -> edge
    - modes=bicycle
    - click start edge, end edge, ENTER



Persons

DEFAULT_PEDTYPE

Internal attributes

id	pf_0
color	yellow
departPos	base
begin	0.00

Flow attributes

terminate	end
spacing	poisson
end	1000.00
rate	0.50

Person plans

personTrip: edge->edge

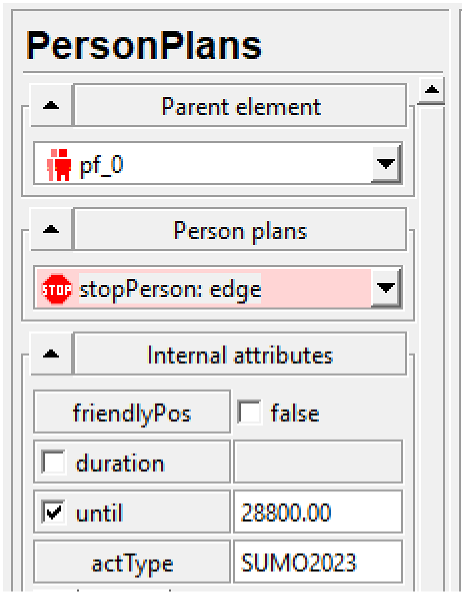
Internal attributes

arrivalPos	
vTypes	
modes	bicycle
lines	ANY

folder: 03_bicycles

Define bicycle flow (3)

- person plan mode (C, for now)
- click on person or select from list
- stopPerson:edge
  - uncheck duration
  - until=8:0:0
  - actType=SUMO2023
  - click on last edge (entrance to the DLR)
- personTrip: edge -> edge
  - modes=bicycle
  - click *final* edge, ENTER (going back to train station)
    - start edge is implicit from previous plan item
- save demand (Ctrl+Shift+D)
- save .sumocfg



PersonPlans

Parent element

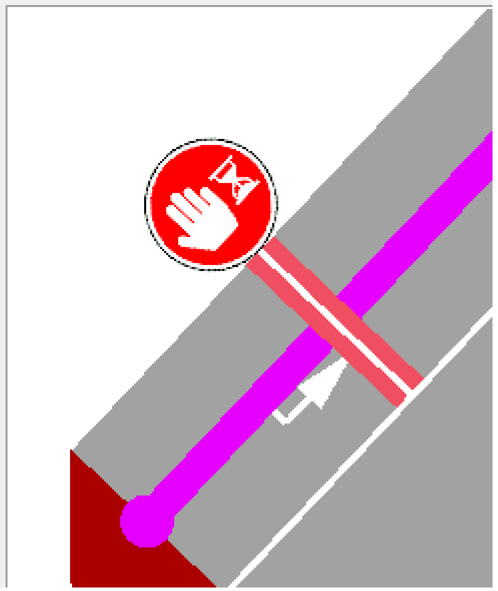
pf_0

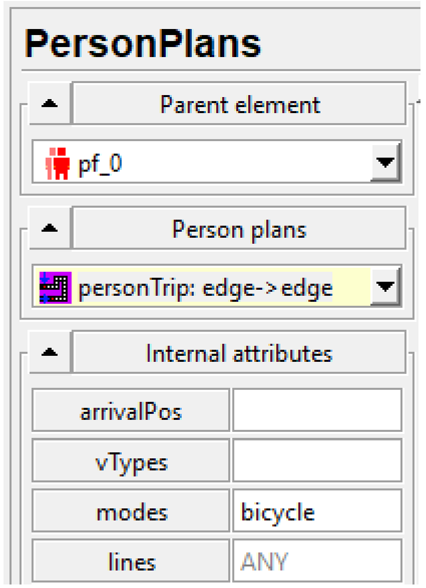
Person plans

stopPerson: edge

Internal attributes

friendlyPos	☐ false
☐ duration	
☑ until	28800.00
actType	SUMO2023





PersonPlans

Parent element

pf_0

Person plans

personTrip: edge->edge

Internal attributes

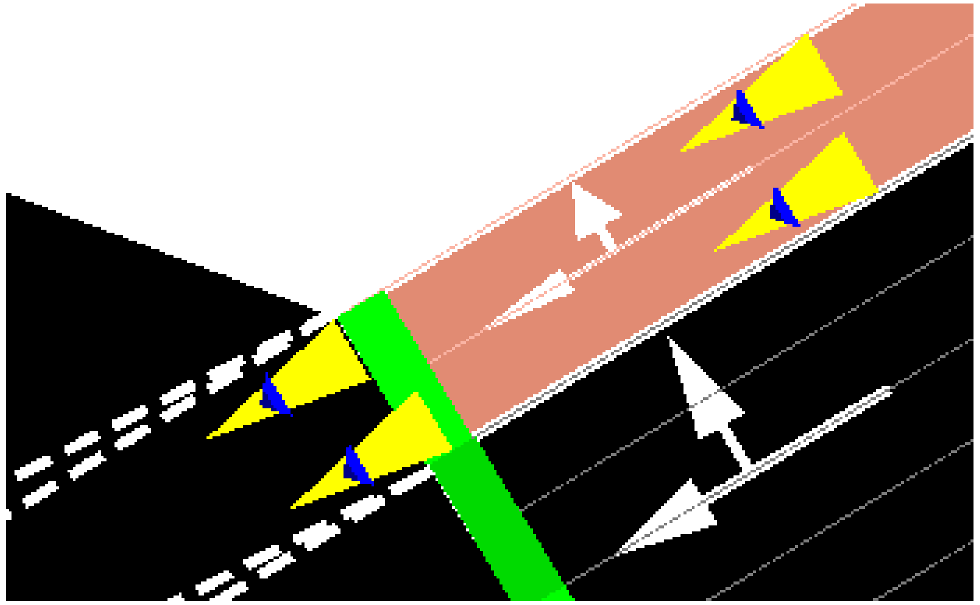
arrivalPos	
vTypes	
modes	bicycle
lines	ANY

folder: 03_bicycles

Evaluating a single scenario run

- define outputs (edit sumocfg with Shift-F10)
  - tripinfo-output: tripinfos.xml
  - personinfo-output: personinfos.xml
  - statistic-output: stats.xml
  - edgedata-output: edgedata.xml

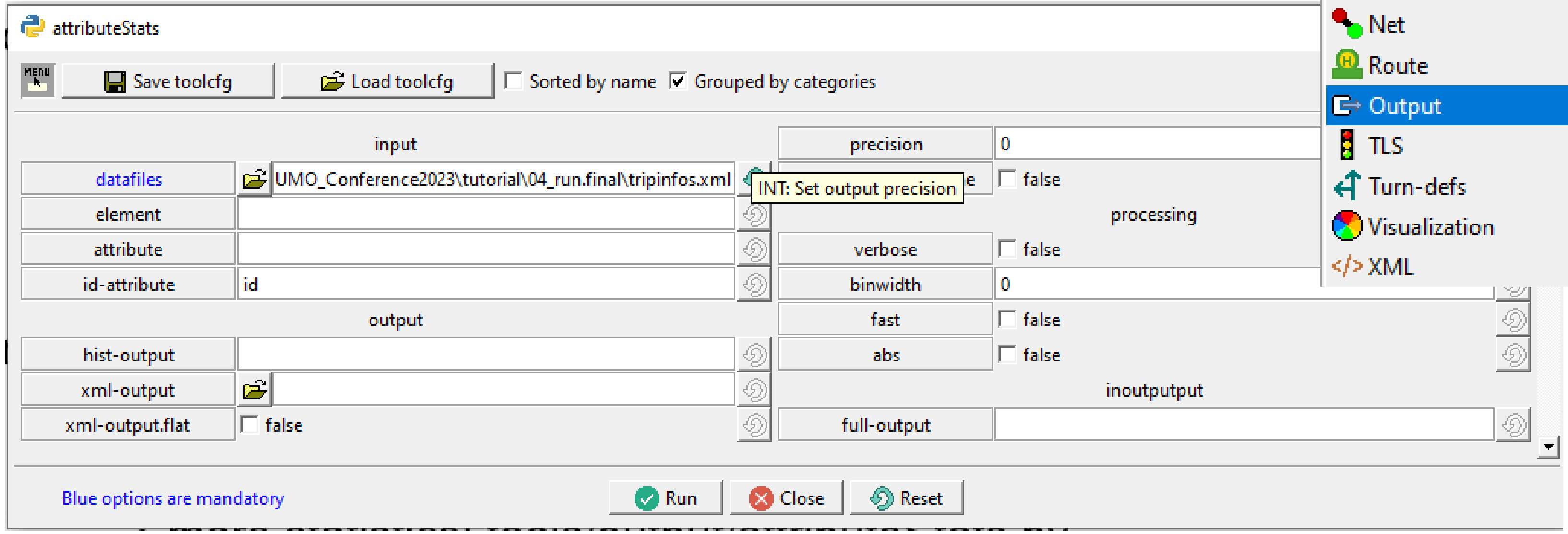
- run the simulation



folder: 04_run

Evaluating a single scenario run

- look at files
  - stats.xml: brief statistical summary of tripinfos.xml and personinfos.xml
  - more statistics: tools/output/attributeStats.py



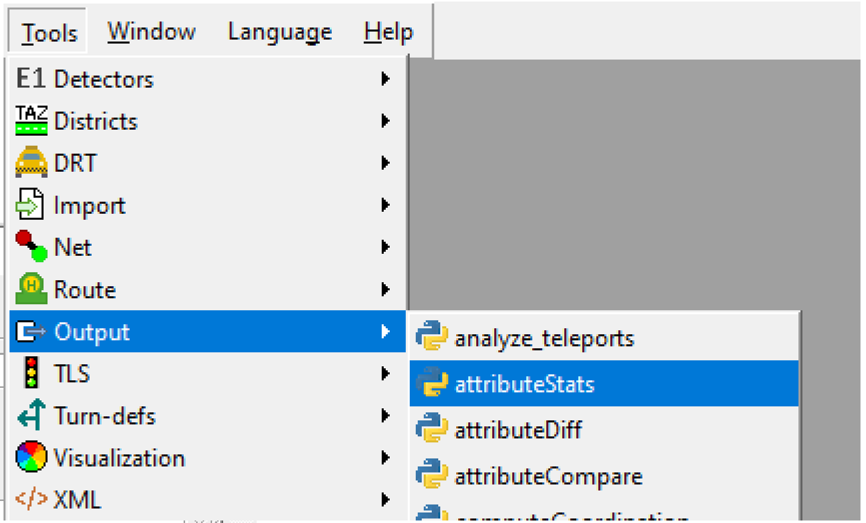
attributeStats

Save toolcfg | Load toolcfg | ☐ Sorted by name | ☑ Grouped by categories

input		precision	0
datafiles	UMO_Conference2023\tutorial\04_run.final\tripinfos.xml		☐ false
element		processing	
attribute		verbose	☐ false
id-attribute	id	binwidth	0
output		fast	☐ false
hist-output		abs	☐ false
xml-output		inoutputput	
xml-output.flat	☐ false	full-output	

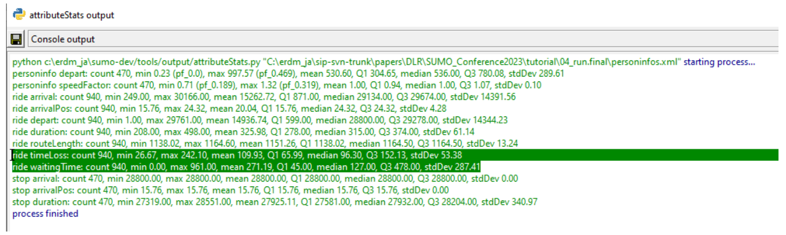
INT: Set output precision

Blue options are mandatory



folder: 04_run

Evaluating a single scenario run



```
python c:\erdm_ja\sumo-dev/tools/output/attributeStats.py "C:\erdm_ja\sip-svn-trunk\papers\DLR\SUMO_Conference2023\tutorial\04_run.final\personinfos.xml" starting process...
personinfo depart: count 470, min 0.23 (pf_0.0), max 997.57 (pf_0.469), mean 530.60, Q1 304.65, median 536.00, Q3 780.08, stdDev 289.61
personinfo speedFactor: count 470, min 0.71 (pf_0.189), max 1.32 (pf_0.319), mean 1.00, Q1 0.94, median 1.00, Q3 1.07, stdDev 0.10
ride arrival: count 940, min 249.00, max 30166.00, mean 15262.72, Q1 871.00, median 29134.00, Q3 29674.00, stdDev 14391.56
ride arrivalPos: count 940, min 15.76, max 24.32, mean 20.04, Q1 15.76, median 24.32, Q3 24.32, stdDev 4.28
ride depart: count 940, min 1.00, max 29761.00, mean 14936.74, Q1 599.00, median 28800.00, Q3 29278.00, stdDev 14344.23
ride duration: count 940, min 208.00, max 498.00, mean 325.98, Q1 278.00, median 315.00, Q3 374.00, stdDev 61.14
ride routeLength: count 940, min 1138.02, max 1164.60, mean 1151.26, Q1 1138.02, median 1164.50, Q3 1164.50, stdDev 13.24
ride timeLoss: count 940, min 26.67, max 242.10, mean 109.93, Q1 65.99, median 96.30, Q3 152.13, stdDev 53.38
ride waitingTime: count 940, min 0.00, max 961.00, mean 271.19, Q1 45.00, median 127.00, Q3 478.00, stdDev 287.41
stop arrival: count 470, min 28800.00, max 28800.00, mean 28800.00, Q1 28800.00, median 28800.00, Q3 28800.00, stdDev 0.00
stop arrivalPos: count 470, min 15.76, max 15.76, mean 15.76, Q1 15.76, median 15.76, Q3 15.76, stdDev 0.00
stop duration: count 470, min 27319.00, max 28551.00, mean 27925.11, Q1 27581.00, median 27932.00, Q3 28204.00, stdDev 340.97
process finished
```

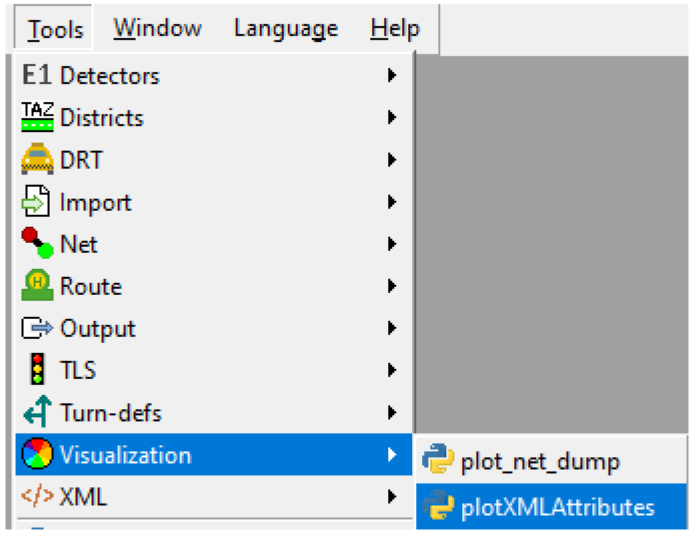
```
tripinfo timeLoss: count 940, min 26.67 (pf_0.443_b0), max 242.10 (pf_0.299_b0),
  mean 109.93, Q1 65.99, median 96.30, Q3 152.13, stdDev 53.38
tripinfo waitingCount: count 940, min 0.00 (pf_0.43_b0), max 9.00 (pf_0.432_b0),
  mean 3.26, Q1 2.00, median 3.00, Q3 4.00, stdDev 1.56
tripinfo waitingTime: count 940, min 0.00 (pf_0.43_b0), max 160.00 (pf_0.414_b0),
  mean 53.14, Q1 18.00, median 41.00, Q3 84.00, stdDev 41.04

ride timeLoss: count 940, min 26.67, max 242.10,
  mean 109.93, Q1 65.99, median 96.30, Q3 152.13, stdDev 53.38
ride waitingTime: count 940, min 0.00, max 961.00,
  mean 271.19, Q1 45.00, median 127.00, Q3 478.00, stdDev 287.41
```

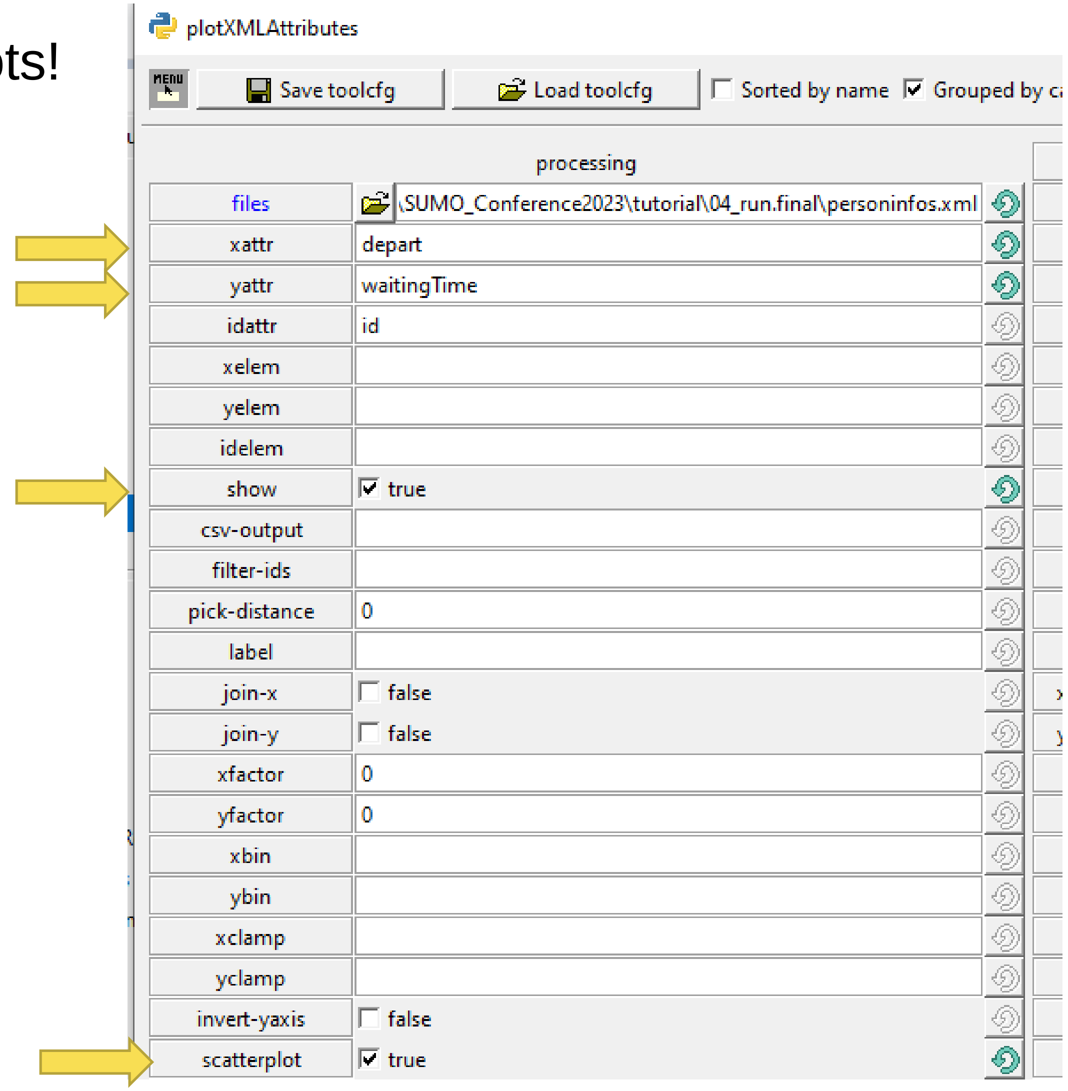
time lost due to slow driving (includes waiting with speed=0)
time spent waiting (speed=0)
time spent waiting for the ride to start

folder: 04_run

Evaluating a single scenario run

- Enough looking at text. Lets have some plots!





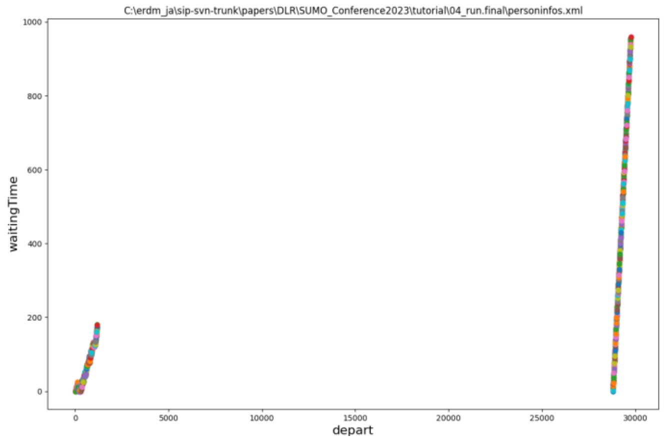
processing

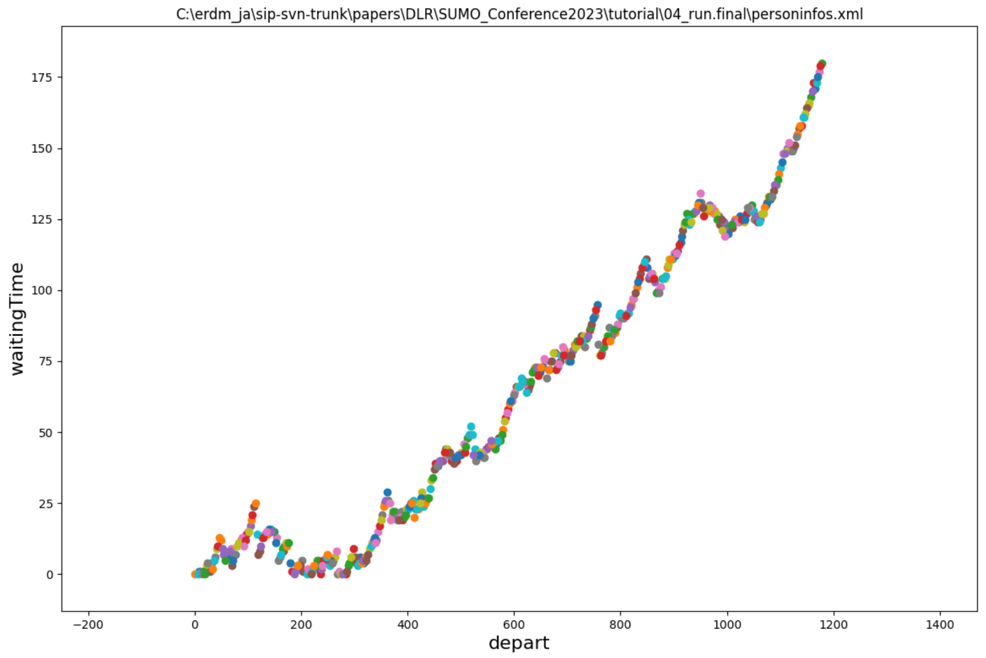
files	\SUMO_Conference2023\tutorial\04_run.final\personinfos.xml
xattr	depart
yattr	waitingTime
idattr	id
xelem	
yelem	
idelem	
show	☑ true
csv-output	
filter-ids	
pick-distance	0
label	
join-x	☐ false
join-y	☐ false
xfactor	0
yfactor	0
xbin	
ybin	
xclamp	
yclamp	
invert-yaxis	☐ false
scatterplot	☑ true

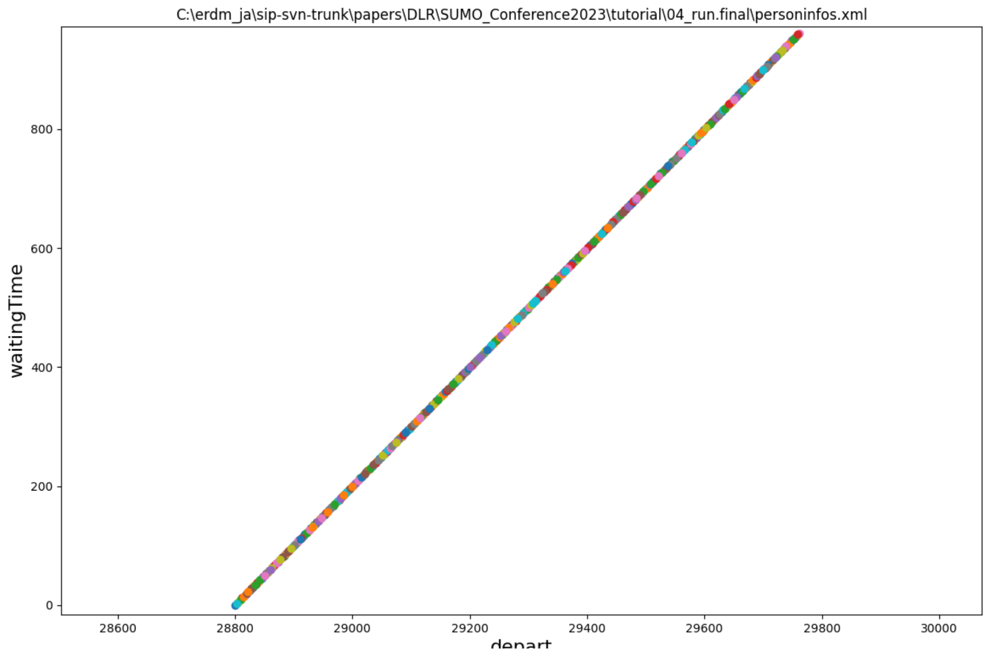
folder: 04_run

Evaluating a single scenario run

- waitingTime of rides: how much is the start of the ride delayed with respect to the desired departure:



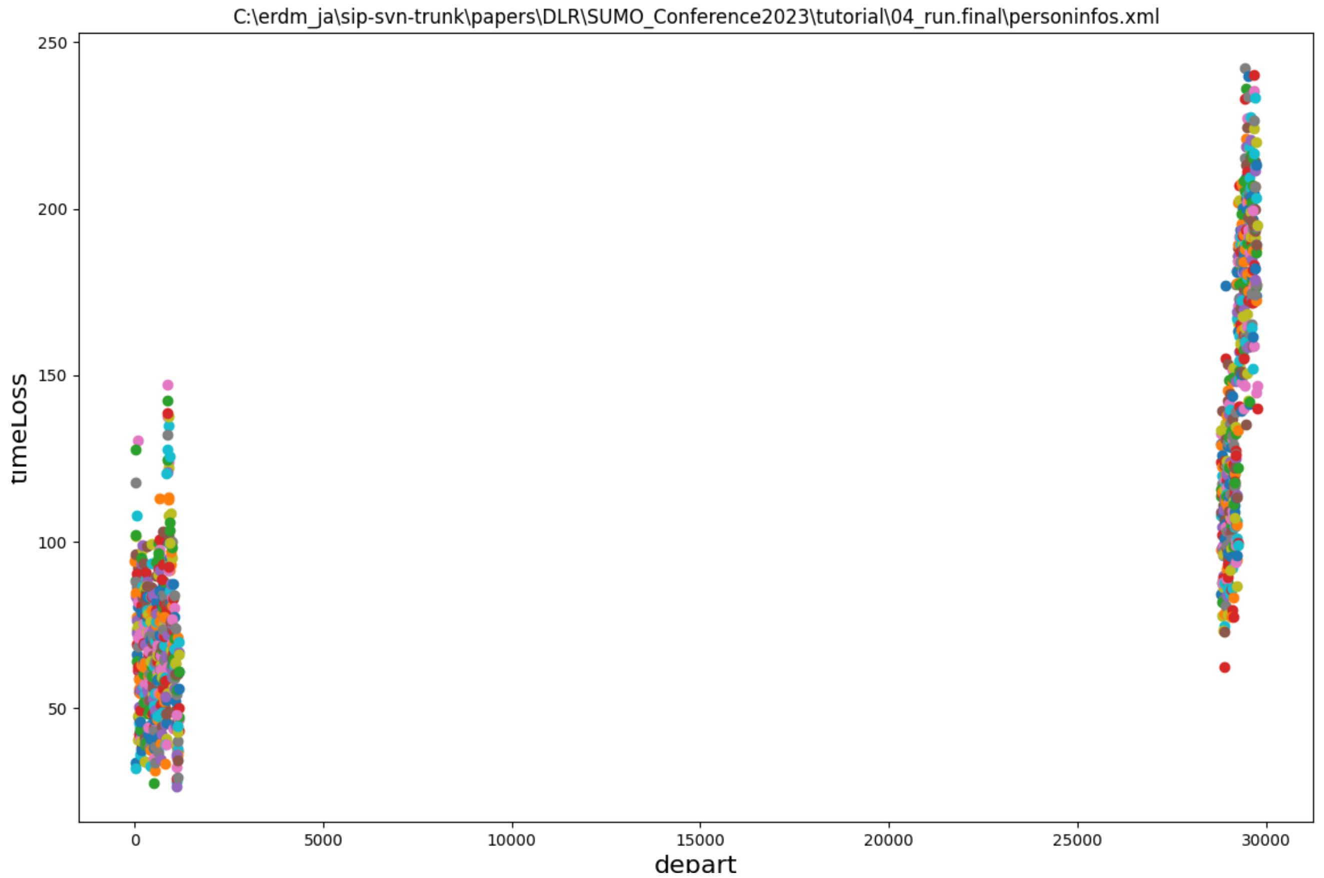




folder: 04_run

Evaluating a single scenario run

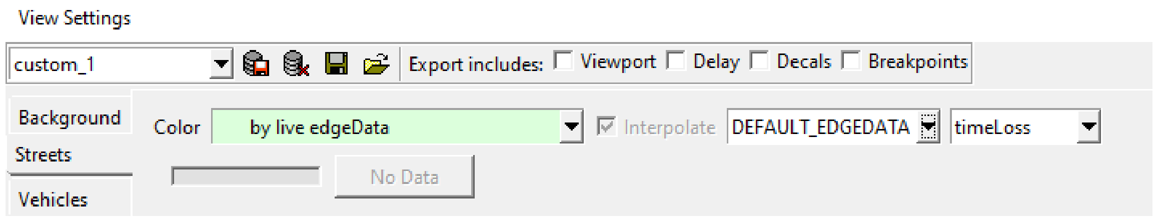
- **timeLoss** of rides: how much time is lost on the road
- the reason for the difference between both legs of travel will surprise you!



folder: 04_run

Evaluating a single scenario run

- plots are nice but you need to look at the simulation
  - visually determine congestion via coloring vehicles "by speed"
  - color edges by accumulated timeLoss
  - color either by 'edgeData' or 'live edgeData'



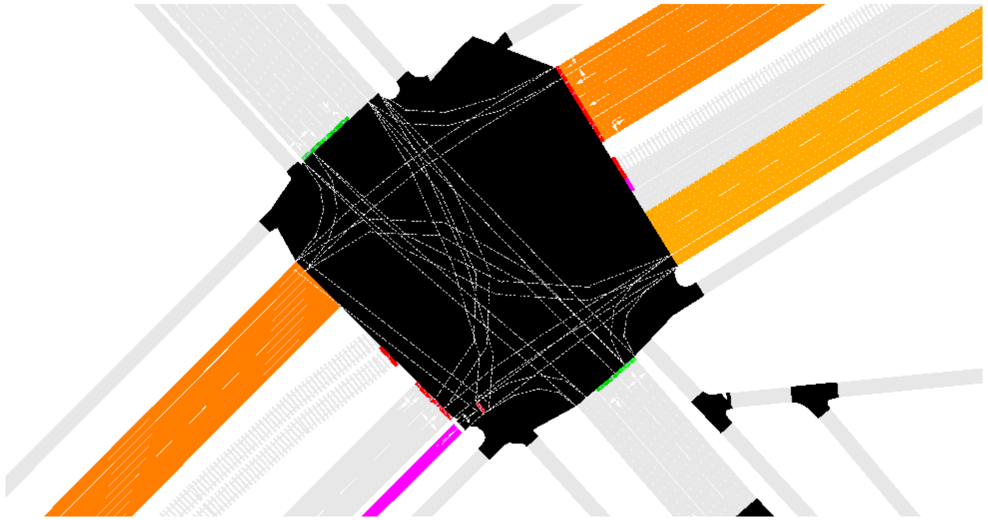
View Settings

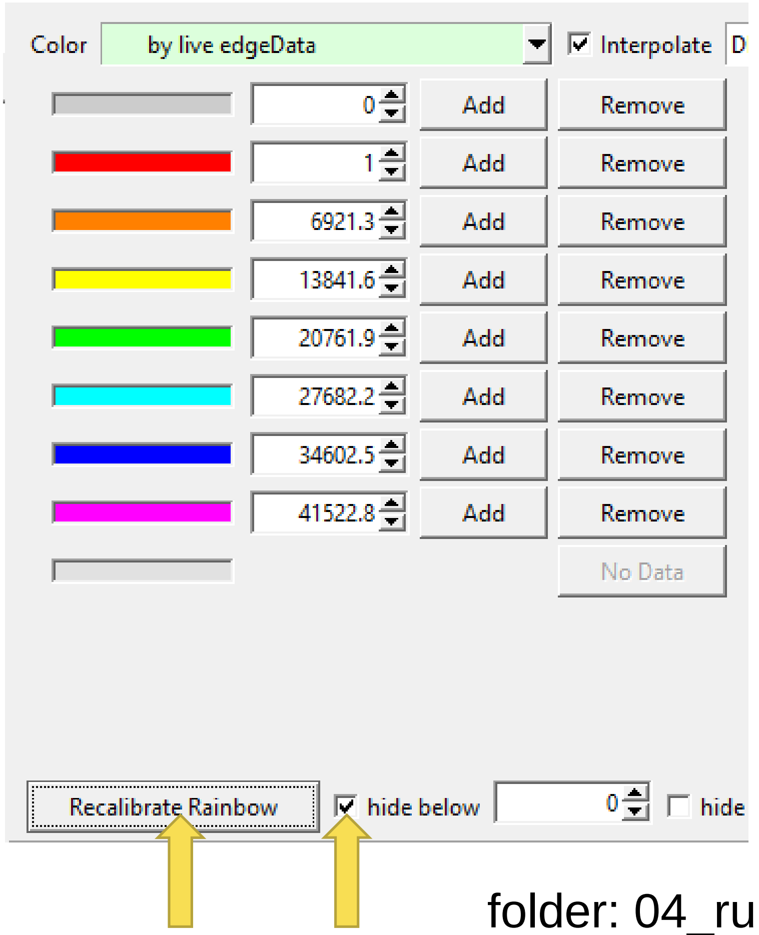
custom_1 | Export includes: ☐ Viewport ☐ Delay ☐ Decals ☐ Breakpoints

Background / Streets / Vehicles

Color: by live edgeData | ☑ Interpolate | DEFAULT_EDGEDATA | timeLoss

No Data





Color: by live edgeData | ☑ Interpolate

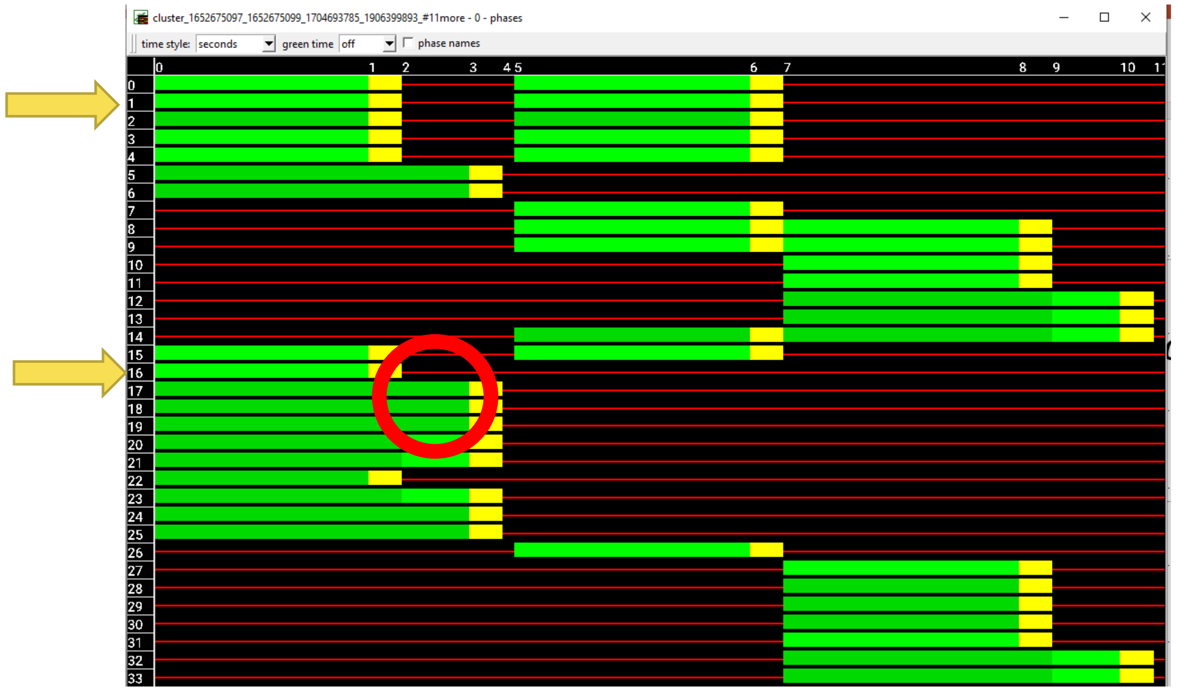
⬜	0	Add	Remove
🟥	1	Add	Remove
🟧	6921.3	Add	Remove
🟨	13841.6	Add	Remove
🟩	20761.9	Add	Remove
🟦	27682.2	Add	Remove
🟦	34602.5	Add	Remove
🟪	41522.8	Add	Remove
⬜			No Data

Recalibrate Rainbow | ☑ hide below 0 | ☐ hide

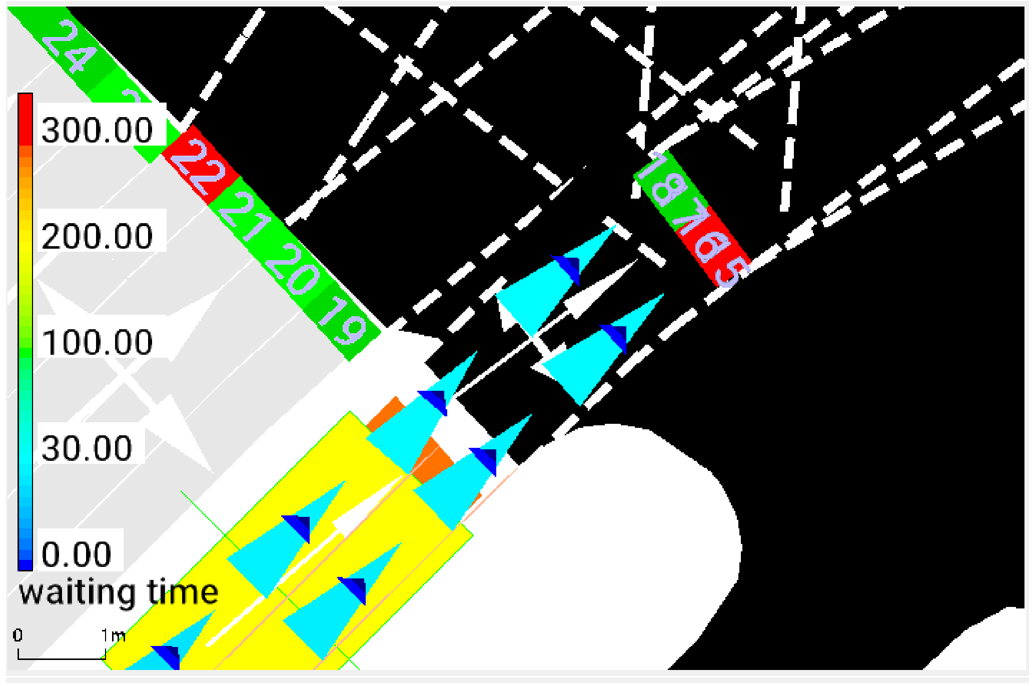
folder: 04_run

Evaluating a single scenario run

- root cause of difference:
  - asymmetrical traffic light signal plans
  - invalid use of actuation detector (remember option ***jam-threshold=30***)


cluster_1652675097_1652675099_1704693785_1906399893_#11more - 0 - phases

time style: seconds green time off phase names

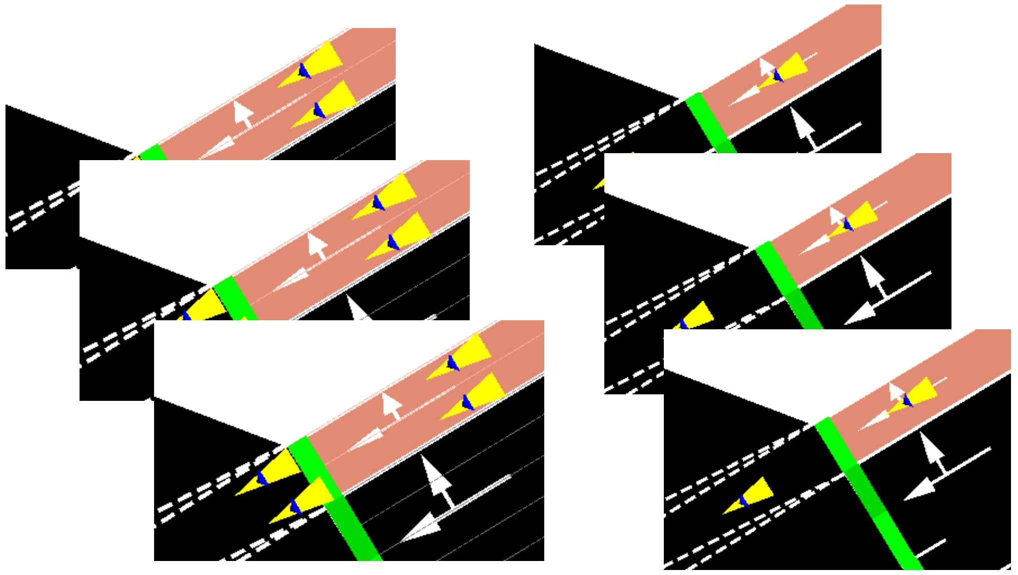


- remember: signal plans are not part of OSM and must be "guessed"

folder: 04_run

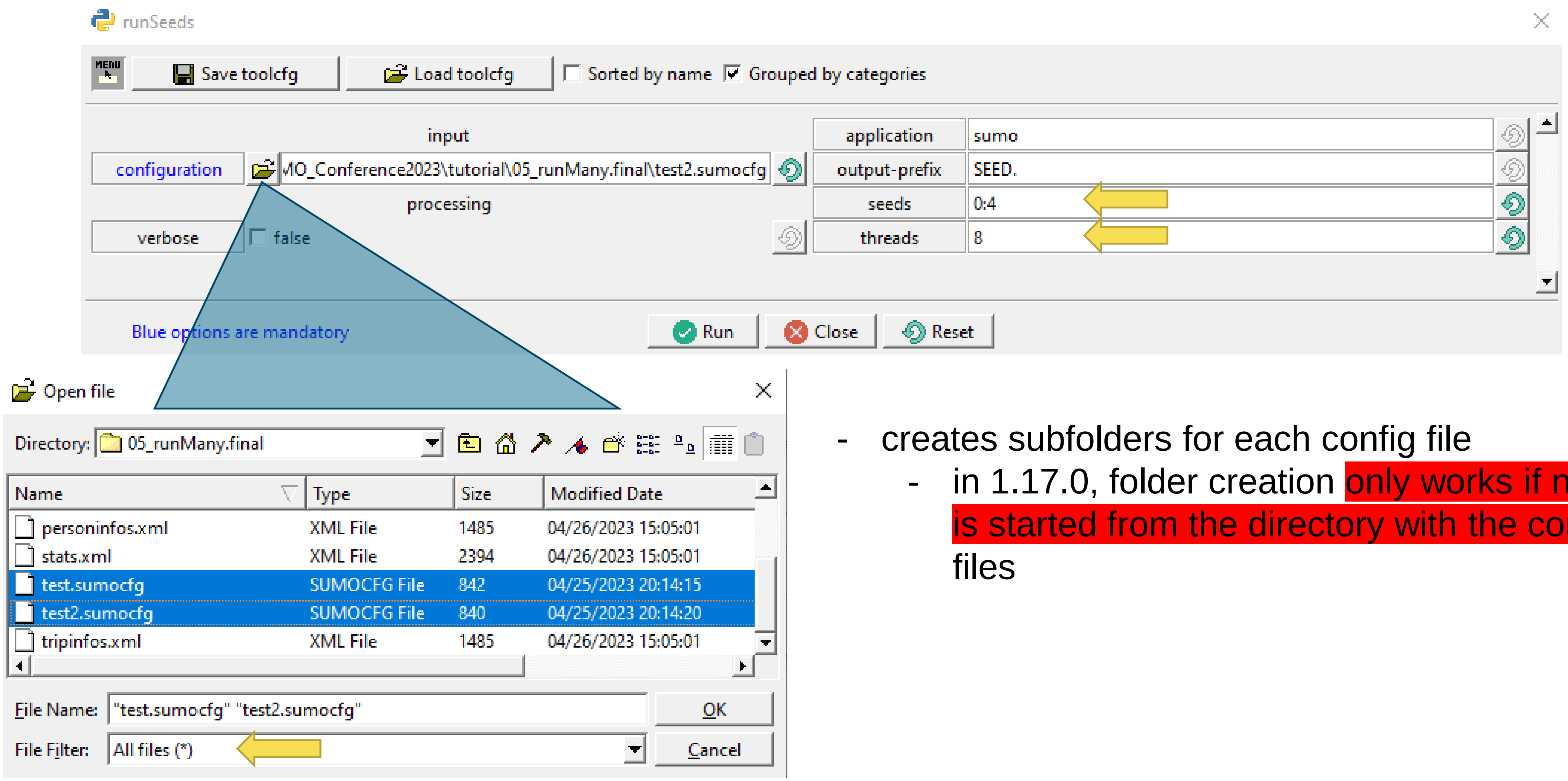
Evaluating multiple scenario runs

- run with different configurations
  - test2.sumocfg with lateral resolution 0
- run with different random seeds
  - look at files
  - look at plots



folder: 05_runMany

runSeeds.py



☐ Sorted by name ☑ Grouped by categories

input

configuration: MO_Conference2023\tutorial\05_runMany.final\test2.sumocfg

processing

verbose: ☐ false

application	sumo
output-prefix	SEED.
seeds	0:4
threads	8

Blue options are mandatory

Directory: 05_runMany.final

Name	Type	Size	Modified Date
personinfos.xml	XML File	1485	04/26/2023 15:05:01
stats.xml	XML File	2394	04/26/2023 15:05:01
test.sumocfg	SUMOCFG File	842	04/25/2023 20:14:15
test2.sumocfg	SUMOCFG File	840	04/25/2023 20:14:20
tripinfos.xml	XML File	1485	04/26/2023 15:05:01

File Name: "test.sumocfg" "test2.sumocfg"

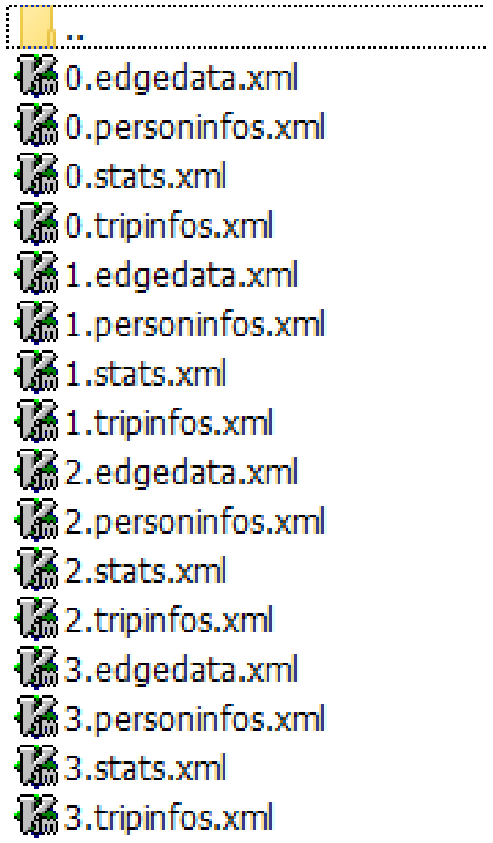
File Filter: All files (*)

- creates subfolders for each config file
  - in 1.17.0, folder creation only works if netedit is started from the directory with the config files

folder: 05_runMany

runSeeds.py

- creates one folder for each .sumocfg
    - option to put everything in one folder is coming
- creates output files prefixed with the random seed
- a simple way to run scenarios in parallel

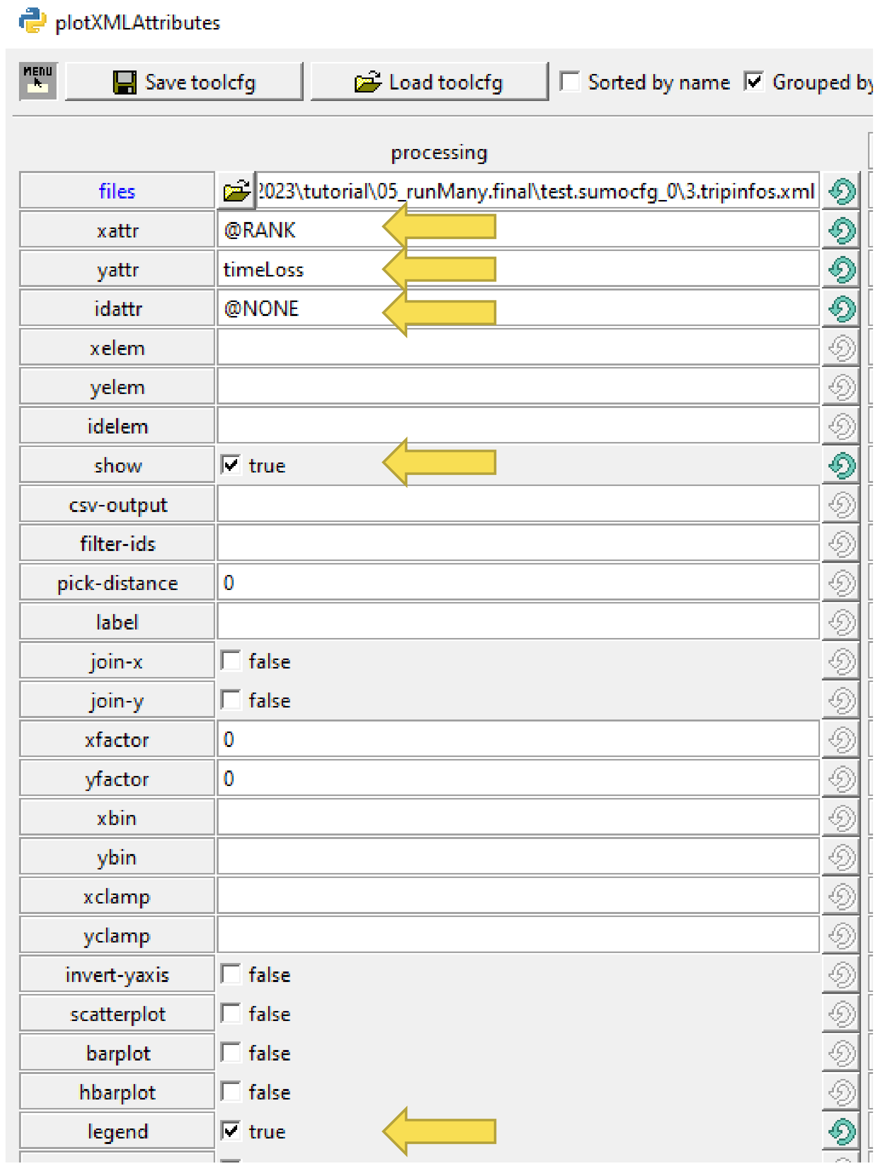


```
..
0.edgedata.xml
0.personinfos.xml
0.stats.xml
0.tripinfos.xml
1.edgedata.xml
1.personinfos.xml
1.stats.xml
1.tripinfos.xml
2.edgedata.xml
2.personinfos.xml
2.stats.xml
2.tripinfos.xml
3.edgedata.xml
3.personinfos.xml
3.stats.xml
3.tripinfos.xml
```

folder: 05_runMany

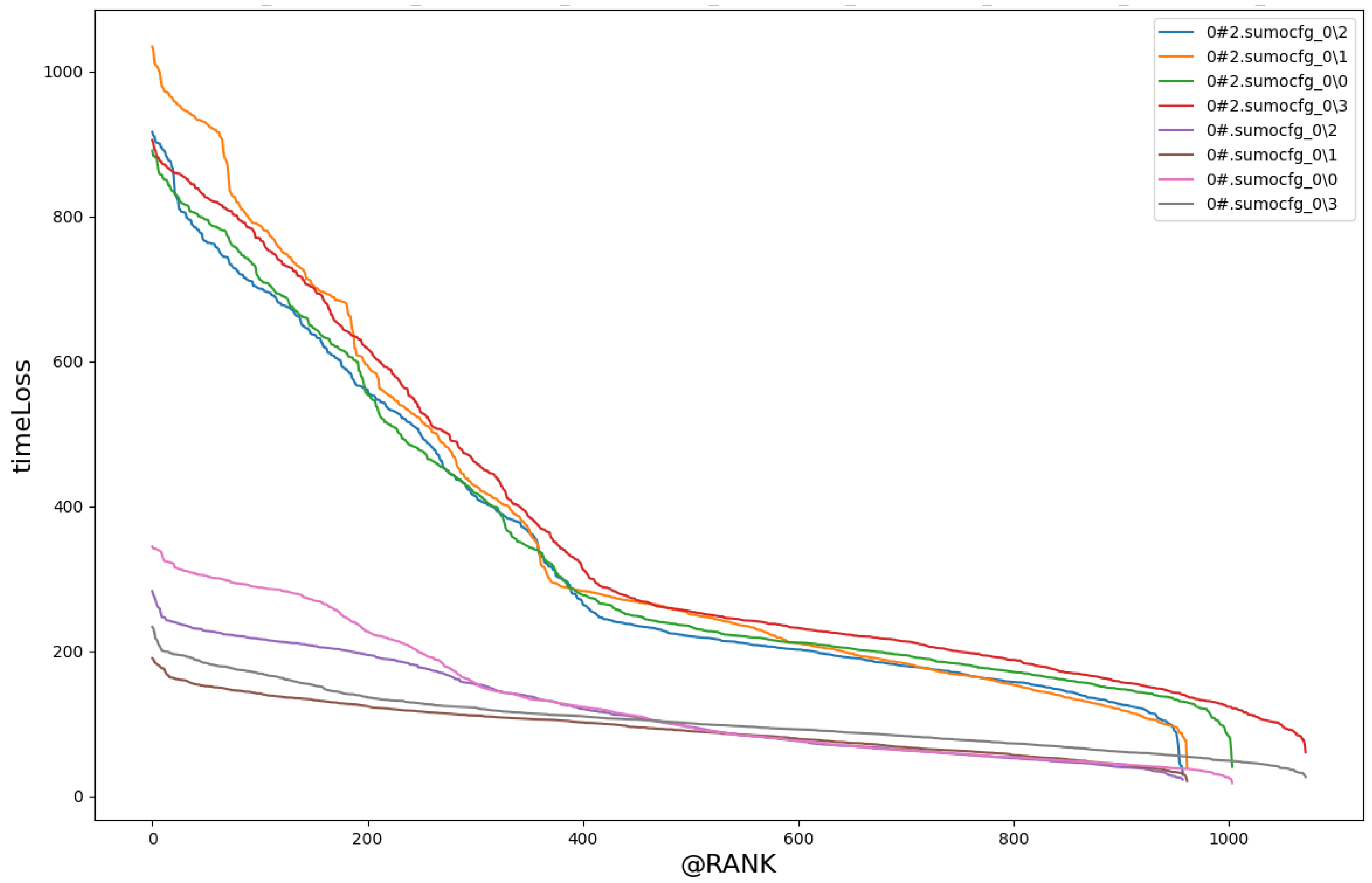
plot timeLoss over all runs

- select all tripinfo files from both folders
  - (copy paste names or files for now)
- xattr=@RANK sorts the y-values uses the sorting rank as x value
- idattr=@NONE means we ignore the vehicle ids and group data points by filename



files	2023\tutorial\05_runMany.final\test.sumocfg_0\3.tripinfos.xml
xattr	@RANK
yattr	timeLoss
idattr	@NONE
xelem	
yelem	
idelem	
show	✓ true
csv-output	
filter-ids	
pick-distance	0
label	
join-x	false
join-y	false
xfactor	0
yfactor	0
xbin	
ybin	
xclamp	
yclamp	
invert-yaxis	false
scatterplot	false
barplot	false
hbarplot	false
legend	✓ true

folder: 05_runMany


0#2.sumocfg_0\2
0#2.sumocfg_0\1
0#2.sumocfg_0\0
0#2.sumocfg_0\3
0#.sumocfg_0\2
0#.sumocfg_0\1
0#.sumocfg_0\0
0#.sumocfg_0\3
timeLoss
@RANK
0
200
400
600
800
1000

Conclusion

- Use tools/osmWebWizard.py to get a quick start
- Read the documentation / FAQ at http://sumo.dlr.de/docs
- Report any bugs you find to sumo-user@eclipse.org
- Share your scenarios and results

- Talks to us. We are always looking for project partners! sumo@dlr.de